UNIVERSIDADE DO MINDELO

DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR

CURSO DE LICENCIATURA EM

ENGENHARIA EM ENERGIAS RENOVÁVEIS

Tema: Estação Meteorológica com Acesso Remoto

Autor: Elízio Cardoso Mendes, N.º 3103

Orientador: Doutor João Dias

Coordenação: João Dias, Ph.D.

Mindelo, 2018

CURSO DE LICENCIATURA EM ENGENHARIA EM ENERGIAS RENOVÁVEIS

RELATÓRIO DE TRABALHO DE CONCLUSÃO DE CURSO

ANO LETIVO 2017/ 2018 – 4º ANO

ESTAÇÃO METEOROLÓGICA COM ACESSO REMOTO

A U T O R: ELÍZIO CARDOSO MENDES, N.º 3103

O R I E N T A D O R: DOUTOR JOÃO DIAS

Coordenação: João Dias, Ph.D.

Mindelo, 2018

Elízio Cardoso Mendes

ESTAÇÃO METEOROLÓGICA COM ACESSO REMOTO

Projeto de trabalho de conclusão do curso, apresentado à Universidade do Mindelo como parte dos requisitos para obtenção do grau de licenciado em Engenharia em Energias Renováveis.

Orientador:

Doutor João Dias

Mindelo, 2018

DEDICATÓRIA

Dedico este projeto a todos que de uma forma ou de outra ajudaram-me durante estes anos de formação, em especial a minha família, os meus professores, os meus colegas e amigos e ao falecido professor, colega e amigo Samuel Delgado.

AGRADECIMENTOS

Começo por agradecer a Deus por iluminar os meus caminhos e por dar-me forças para que eu siga sempre em frente e conseguir alcançar os meus objetivos.

De igual modo, agradeço aos meus familiares, em especial aos meus pais pelo grande apoio que deram-me e continuam a dar, pois sem eles não tinha como chegar onde estou, e aos meus irmãos por estarem sempre presentes.

Ao orientador e coordenador do curso João Dias pelo auxílio, dedicação e paciência que foram determinantes para conclusão do projeto.

Aos colegas de curso, amigos feitos na Universidade do Mindelo, desde o primeiro ano até o último pelos grandes momentos de convivência, pois foram essenciais para o encerramento deste ciclo.

Um especial obrigado ao Engº Aníbal Mota pelo ensinamento, pelo encorajamento em seguir com esse projeto e pela paciência que teve comigo durante o tempo que esteve presente.

Por fim agradeço a todos que de alguma forma, direta ou indiretamente, contribuíram para o meu sucesso durante este percurso.

“Não creio que haja uma emoção, mais intensa para um inventor do que ver suas criações funcionando. Essa emoção faz você esquecer de comer, de dormir, de tudo.”

Nikola Tesla

"Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível."

Charles Chaplin

RESUMO

Neste projeto desenvolveu-se uma estação meteorológica com acesso remoto usando a plataforma Arduíno Uno e o microcomputador Raspberry Pi em conjunto com sensores que medem seis variáveis climáticas: temperatura, humidade, pressão, altitude, intensidade luminosa e velocidade de vento. Os dados recolhidos por estes sensores são armazenados num banco de dados e posteriormente tornados acessíveis a dispositivos remotos através de um *website*. Para o projeto, o *website* é considerado como sistema de monitoramento. Este *website* é responsável pela exibição dessas variáveis climáticas em tempo real, através de medidores, gráficos de variação e tabela de registo de dados.

Fez-se uma descrição de cada componente escolhido, onde foi realizado um estudo comparativo que avaliou as características de eventuais componentes alternativos. As instruções para a montagem e funcionamento de todo o sistema são descritas ao longo do projeto.

Palavras-chave: Estação Meteorológica, Sensores, Arduino Uno, Raspberry Pi, *Website*

ABSTRACT

In this project, a weather station with remote access was developed using the Arduino Uno platform and the Raspberry Pi microcomputer together with sensors that measure six climatic variables: temperature, humidity, pressure, altitude, light intensity and wind speed. The data collected by these sensors are stored in a database and then made accessible to remote devices through a website. For the project, the website is considered as a monitoring system. This website is responsible for displaying these climate variables in real time, through meters, variation graphs and data logging tables.

A description of each component was made, where a comparative study was carried out to evaluate the characteristics of eventual alternatives. Instructions for the assembly and operation of the entire system are described throughout the project.

Keywords: Weather Station, Sensors, Arduino Uno, Raspberry Pi, Website

ÍNDICE

ÍNDICE DE FIGURA

ÍNDICE DE TABELA

LISTA DE SIGLAS E ABREVIATURAS

A	*Ampere*
ANSI	*American National Standard Institute*
CI	Circuito Integrado
CPU	*Central Processing Unit* - Unidade Central de Processamento
EEPROM	*Electrically-Erasable Programmable Read-Only Memory*
GND	*Ground*
GPIO	*General Purpose Input/Output*
HDMI	*High-Definition Multimedia Interface*
I2C	*Inter-Integrated Circuit*
IDE	*Integrated Development Environment*
LDR	*Light Dependent Resistor*
LED	*Light Emitting Diode*
PC	*Personal Computer*
PWM	*Pulse-Width Modulation*
RAM	*Random Access Memory*
RPI 3B	Raspberry Pi 3 modelo B
RPI	Raspberry Pi
SD	*Secure Digital*
SGBD	Sistema de Gerenciamento de Banco de Dados
SPI	*Serial Peripheral Interface*
SQL	*Structured Query Language*
SSH	*Secure Shell*
UART	*Universal Asynchronous Receiver/Transmitter*
URL	*Uniform Resource Locator*
USB	*Universal Serial Bus*
V	*Volt*
VGA	*Video Graphics Array*
Wi-Fi	*Wireless Fidelity*
WLAN	*Wireless Local Area Network*

I. INTRODUÇÃO

Todos interessamo-nos pelo clima e, em vários países, boletins de previsão do tempo são programas que passam com frequência na televisão. Pessoas em todo o mundo têm a necessidade de saber a condição do tempo, do presente ou do futuro, para que possam saber as possibilidades de semeadura, plantação e/ou colheita (no caso de atividades ligadas à agricultura), condições de viagens (marítimas, aéreas ou terrestre), previsão de possíveis ocorrências naturais (ciências ambientais) tais como furacões, saber as condições para a prática de desportos, ou simplesmente o que vestir ou se é necessário levar um guarda-chuva. Desta forma o projeto poderá fornecer serviços confiáveis e efetivos para apoiar a segurança de propriedades e vidas, bem como o bem-estar geral da população.

Esse projeto poderá também ser aplicado para monitorar as condições meteorológicas em curtos intervalos de tempo, com objetivos imediatos. Entre eles, o de verificar a ocorrência de ventos fortes, temperaturas médias, mínimas e máximas diárias, variações na intensidade e direção dos ventos, pressão atmosférica e humidade. Poderá ser aplicada nas áreas das energias renováveis em que, para além de ser um projeto energicamente eficiente, será também de grande utilidade no estudo das condições do vento de um determinado local para implementação de turbinas eólicas, por exemplo.

O presente projeto apresenta uma forma de adquirir informações mais detalhada possível de uma estação meteorológica, através do acesso remoto. A internet servirá como base principal para o acesso a essas informações obtidas pelo sistema. Essas informações poderão ser utilizadas para vários fins, nomeadamente o desenvolvimento de páginas que fornecem as condições climáticas de um determinado local.

Refletindo de uma forma diferente, propõe-se neste projeto, soluções mais avançadas, com sistema microprogramável através de microcontrolador e microcomputador, com sensores de baixo custo, mas de elevada qualidade, e acesso às informações remotamente.

Pretende-se que o protótipo tenha aplicabilidade a pequenas estações, pois o desenvolvimento para estações de grande porte depende de sensores com precisão muito maior, exatidão dos dados e cálculos estatísticos.

1.1 Objetivos

O objetivo principal deste projeto é desenvolver, construir e testar um protótipo de uma estação meteorológica com controlo remoto dotado de sensores de temperatura, humidade, pressão atmosférica, altitude, intensidade luminosa e velocidade de vento, assente num microcontrolador Arduíno Uno que por sua vez estará ligado a um microcomputador Raspberry Pi.

Terá como propósito, recolher dados para análise do clima para, posteriormente esses dados serem utilizados para previsão do tempo, monitorização e caracterização do clima de um determinado local. Permitir ainda, que empresas (exemplo: aviação, marítima, agrícola, etc.) possuam um dispositivo energeticamente eficiente fornecendo serviços confiáveis e efetivos para apoiar a segurança de propriedades e vidas, bem como o bem-estar geral da população.

1.2 Metodologia

Buscando analisar a temática proposta, como em qualquer projecto, começou-se por fazer um levantamento do estado da arte a fim de identificar as tecnologias e aborgens existentes, ou seja, uma pesquisa maioritariamente de fontes secundárias sobre o tema Estação Meteorológica com Acesso Remoto, desenvolvida a partir de materiais publicadas tais como: livros, artigos, principalmente em dissertações e teses obtidas na internet em formatos PDF, para dar sustentabilidade a pesquisa e para ajudar a compreender os principais conceitos.

Inicialmente pretende-se estudar todos os componentes do projeto, desde sensores até o microcontrolador Arduíno Uno e o microcomputador Raspberry Pi. Posteriormente estuda-se a criação de um banco de dados, desenvolvendo o código para converter dados dos sensores em informações. Em seguida saber como a estação meteorológica pode partilhar informações com visualizadores através de um *website* simples.

Para finalizar pretende-se elaborar o protótipo, implementando o sistema e elaborar testes para uma melhor avaliação das funcionalidades do sistema, fazer os reajustes necessários e, no fim, realizar a conclusão do relatório final.

1.3 Estrutura do Projeto

De acordo com os objetivos definidos anteriormente, o projeto encontra-se estruturado em seis capítulos.

No primeiro capítulo, fez-se o enquadramento do referente tema do projeto.

No segundo, foi reservado para uma breve introdução teórica sobre o conceito de estações meteorológicas e apresentação das variáveis climáticas que serão medidas pela estação.

No terceiro capítulo, estudou-se o microcontrolador Arduíno Uno, o microcomputador Raspberry Pi 3 modelo B e os sensores que serão utilizadas ao longo do projeto.

No quarto capítulo, fez-se uma abordagem sobre as ferramentas e tecnologia utilizadas no desenvolvimento do projeto que servirão para armazenar e partilhar em tempo real os dados obtidos através de um *website* de modo a ser acessível por outros dispositivos.

No quinto e penúltimo capítulo, apresenta-se a implementação do projeto, no que diz respeito às configuraçoes feitas no RPI, a implementação do sistema de registo de dados, o monitoramento e gerenciamento do mesmo e, por fim, a parte da interligação do circuito eletrónico.

E para finalizar, o sexto capítulo, apresenta-se a conclusão geral do projeto, incluindo a análise do projeto desenvolvido e futuros trabalhos que poderão ser baseados a partir desde projeto.

II. ESTAÇÕES METEREOLÓGICAS

2.1 Conceitos e Aplicações

Algumas áreas de negócios dependem de uma boa condição climática para obterem sucesso e, como exemplo, podemos citar o ramo do agronegócio, construção civil, aviação, pesca ou espectáculos ao ar livre, visto que quando acontece uma mudança brusca de tempo, muitas vezes a vida das pessoas cujam actividade profissional estejam dependentes do clima nessas áreas pode ser colocada em risco.

As estações meteorológicas são equipamentos utilizados em todo o mundo como ferramentas que realizam registo dos fenómenos climáticos, utilizando sensores para análise de parâmetros meteorológicos. Essas estações recolhem e analisam os dados, e desse modo é definido o clima de um determinado local ou região.

Entre as informações geradas com os registos das estações meteorológicas estão: a velocidade média de vento, temperaturas médias e máximas diárias, direção do vento, humidade relativa do ar, altitude, pressão atmosférica, volume de chuva e outras mais.

Vale destacar que a finalidade de uma estação meteorológica não é de realizar previsão do tempo (apesar de algumas fazerem), e sim possibilitar um monitoramento das variáveis climáticas.

Segundo o blogue Mundoclima (autor desconhecido), 2017, de acordo com o tipo de equipamentos que possuem, as estações meteorológicas podem ser classificadas em automáticas e convencionais.

Nas estações meteorológicas automáticas são utilizadas sensores eletrónicos para a mediação de inúmeras variáveis, entre elas a temperatura, humidade relativa do ar, direção e intensidade do vento, precipitação pluviométrica, pressão atmosférica, altura de nuvens (até aos 1500 metros), entre outras. Todos os dados são enviados para uma central, onde programas computacionais integram todas as informações e permitem uma análise das condições climáticas.

Nestas estações, as informações meteorológicas são adquiridas de, minuto em minuto e, a cada hora, estes dados são integrados e disponibilizados para serem transmitidos para uma central de meteorologia, via *wireless* (geralmente por ondas de rádio), GPRS (General Packet Radio Service) ou satélite. [Mundoclima, 2017]

No caso das estações meteorológicas convencionais, o monitoramento é feito presencialmente e periodicamente por técnicos que recolhem os dados, por meio de instrumentos dispostos numa determinada área, sistematizando-os e criando uma base de dados, com informações referentes a temperatura, ao volume de chuva, ao tempo de insolação, a pressão, a evaporação, entre outras variáveis.

As medições são realizadas não mais de quatro vezes por dia, impreterivelmente às 00, 06, 12 e 18 horas. Neste tipo de estação meteorológica, as observações também são visuais, como o tipo de nuvem e a visibilidade. Ou seja, nos horários de observação o técnico avalia, quantifica e classifica as nuvens daquele momento. [Mundoclima, 2017]

A utilização dos dados das estações meteorológicas acima referenciadas são destinadas a várias finalidades e, o seu uso, torna-se essencial para o sucesso e desempenho de muitas atividades que dependem desses dados, nomeadamente na Pesca, Agricultura e Aeronaútica:

a) **Pesca -** numa condição desfavorável ou mesmo que represente perigo, ter informações sobre condições climáticas pode oferecer segurança aos trabalhadores do mar. Ou seja, para embarcações menores é preciso verificar a velocidade do vento e a ocorrência de chuvas, uma vez que em mar aberto a navegação pode correr perigo. Além disso, muitas espécies de peixes dependem de certas condições climáticas, o que ajuda o pescador a verificar tais informações através dos dados meteorológicos.

b) **Agricultura -** para o produtor, analisar e acompanhar os dados climáticos possibilitará o planeamento das épocas de plantio, da colheita e produtividade, de forma que períodos de maior chuva ou de extremo calor não possam prejudicar a atividade.

c) **Aeronáutica -** a previsão de tempo possibilita conhecer em que condições as áreas de pouso e descolagem estão e, consequentemente, precaver-se de situações inesperadas ou

que coloquem em risco as operações. Além disso, é possível realizar um planeamento antecipado de rotas para o caso de condições inapropriadas ao voo.

2.2 Variáveis Climáticas Medidas

As variáveis escolhidas para medição da estação meteorológica foram:

- **Temperatura** - Medição da temperatura em graus Celsius (°C).
- **Humidade Relativa do Ar -** Medição da humidade relativa do ar - de modo indireto - em percentagem (%).
- **Pressão Atmosférica -** Medição da pressão atmosférica em Pascal (Pa) ou hectopascal (hPa).
- **Altitude –** Unidade de medida em metro (m).
- **Intensidade Luminosa -** Unidade de medida em Lux (lux).
- **Velocidade de Vento –** Medição da velocidade de vento em metro por segundo (m/s).

Essas variáveis foram escolhidas por serem as mais relevantes de uma estação meteorológica, sendo que muitas análises e previsões podem ser feito pela comparação destas medidas ao longo de determinados períodos de tempo.

É de referir que o projeto não contará com o sensor de direção de vento porque a sua implementação apresentou um alto custo. Mas como podemos observar, é possível adicionar mais sensores, para além da direção do vento, o sensor de precipitação e outros.

III. ESTUDO DOS COMPONENTES

Este capítulo é destinado exclusivamente ao estudo dos componentes eletrónicos utilizados ao longo do projeto e que só estão disponíveis no mercado exterior. Serão apresentados os componentes, de forma separada, para uma análise e uma descrição técnica dos mesmos.

Os dois primeiros componentes estudados podem ser considerados o "cérebro do projeto". Apresenta-se o microcontrolador e o microcomputador, responsáveis por receber as informações dos sensores e processar essas informações. Finalizando este capítulo, são estudados todos os sensores meteorológicos utilizadas na estação meteorológica.

Nota: As escolhas e aquisição dos componentes foram feitas após uma pesquisa dos principais instrumentos disponíveis no mercado de Estados Unidos de América através da loja *online* Amazon.

3.1 Estudo dos Microcontroladores

A evolução da microeletrónica levou o homem à criação de dispositivos cada vez menores e mais repletos de funções programáveis, ou seja, capazes de entender códigos carregados de conjunto de instruções. Os dispositivos que descodificam esses códigos são denominados de microprocessadores.

Um microprocessador incorpora num único circuito integrado (CI) as funções de uma Unidade Central de Processamento (CPU). São dispositivos programáveis que recebem dados digitais, capazes de descodificar instruções, processá-las e fornecer resultados com saída, ativando registos ou outros mecanismos. Além disso, pode-se ler e escrever em memórias e realizar operações lógicas e aritméticas.

Portanto, pode-se definir os microcontroladores como sendo equipamentos que usam um microprocessador para aplicações diversas na engenharia. Por outras palavras, microcontroladores são circuitos digitais programáveis, compostos por um microprocessador, memórias e periféricos de entrada e saída, acoplados em um único CI.

Os microcontroladores podem ser usados para controlar os mais diversos equipamentos, desde controlo de LED's até motores elétricos, porque contam com equipamentos periféricos capazes de gerar conversões analógicos digitais (A/D) e modulação de largura de pulso (PWM). De entre os diversos tipos de microcontroladores, escolheu-se para utilização no projeto, o Arduíno Uno, por atender perfeitamente a interface com os sensores escolhidos para o projeto, assim como a aquisição e transmissão dos dados fornecidos.

Ainda entre os principais motivos da escolha do microcontrolador Arduíno Uno destacam-se:

- Possui *hardware* mínimo apropriado para o projeto;
- Alta capacidade de processamento de sinais;
- Quantidade de pinos de dados adequada;
- Facilidade de conexão com os outros componentes;
- Disponibilidade do mercado exterior a baixo custo.

3.1.1 Arduíno Uno Rev3

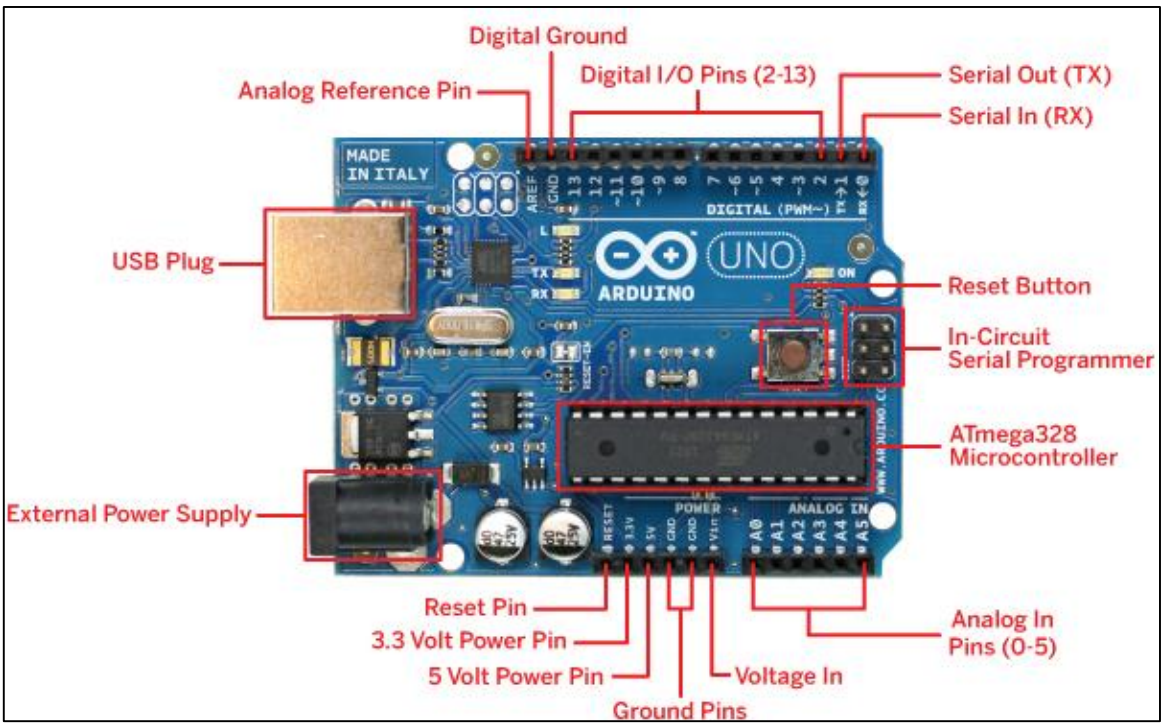
Arduíno é uma plataforma de eletrónica aberta criada em 2005 pelo italiano Massimo Banzi (e outros colaboradores), para a criação de protótipos baseada em *software* e *hardware* livres, flexíveis e de uso simples. O objetivo principal foi de criar uma plataforma de baixo custo, para que os estudantes pudessem desenvolver seus protótipos com o menor custo possível. Contudo, a sua utilização foi alargada para artistas, *designers*, engenheiros e/ou qualquer outra pessoa interessada em criar projetos ou ambientes interativos, uma vez que a sua utilidade aplica-se sobretudo a diversas áreas de atuação para o controlo de sistemas, podendo ter aplicações na área de impressão 3D, robótica, engenharia de transportes, engenharia agronômica, musical, etc.

Essa plataforma é formada por dois componentes principais: o ***Hardware*** (a parte física) e o ***Software*** (o ambiente de desenvolvimento – Arduíno IDE).

O *hardware* é composto por uma placa, formada por um conjunto de componentes eletrónicos, na qual são construídos os projetos. Existem diversas placas oficiais de Arduíno, mas para o projeto escolheu-se o Arduino Uno, por cumprir com todos os requisitos e pela sua disponibilidade no mercado a um custo acessível.

Arduíno Uno é uma placa que contém um microcontrolador baseada no ATmega328P, 14 pinos de entradas/saídas digitais, dos quais 6 podem ser utilizados como saídas PWM, 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB para programação, entrada para alimentação externa, um terminal ICSP (*In-Circuit Serial Programming*) e um botão de reinicialização (*reset*).

Figura 1: Legenda de Pinos do Arduíno Uno Rev3



Fonte: http://bodgarage.repofy.com/?p=959

O componente principal da placa Arduíno Uno Rev3 é o microcontrolador ATMEL ATMEGA328P, um dispositivo de 8 bits da família AVR com arquitetura RISC avançada e com encapsulamento DIP28. Ele conta com 32 KB de memória *flash*, onde 512 bytes são utilizados para o boot loader, 2 KB de RAM e 1 KB de EEPROM. Pode operar até 20 MHz, porém na placa Arduíno Uno trabalha em 16 MHz, valor do cristal externo que está conectado aos pinos 9 e 10 do microcontrolador.

O Arduíno pode obter dados do ambiente, através dos seus pinos de entrada, conectando-o a uma variedade de sensores disponíveis, como também pode atuar para o ambiente, controlando desde iluminações até os motores e e outros periféricos.

Os projetos desenvolvidos com o Arduíno podem ser executados de forma autónoma (sem necessidade de estarem conectados a um computador) ou também podem comunicar-se com um computador para a realização de tarefas, com uso de *software* específico.

A) Alimentação

O Arduíno Uno pode ser alimentado pela conexão USB ou por uma fonte de alimentação externa. A alimentação externa é feita através do conector Jack com polo positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites 6 a 20 V. Porém, se for alimentada com uma tensão abaixo de 7 V, a tensão de funcionamento da placa, que no Arduíno Uno é 5 V, pode ficar instável e quando for alimentada com tensão acima de 12 V, o regulador de tensão da placa pode sobreaquecer e danificar a placa. Dessa forma, é recomendado para tensões de fonte externa valores de 7 a 12 Volts.

No caso em que o cabo USB for conectado a um computador, por exemplo, a tensão não precisa ser estabilizada pelo regulador de tensão porque dessa forma a placa é alimentada diretamente através do cabo USB. O circuito apresenta alguns componentes que protegem a porta USB do computador em caso de alguma anomalia.

A placa não possui botão liga/desliga, ou seja, se desejarmos desligar a alimentação, deve-se retirá-la da alimentação, seja ela alimentada pelo cabo USB ou pelo conector Jack.

Os pinos de alimentação são:

- **Vin:** entrada de alimentação para a placa Arduíno quando uma fonte externa for utilizada. Pode fornecer alimentação por este pino ou, se usar o conector de alimentação, aceder a alimentação por este pino;
- **5V:** a fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino Vin através de um regulador *on-board* ou ser fornecida pela porta USB;
- **3V3:** alimentação de 3,3 V fornecida pelo circuito integrado FTDI (controlador USB). A corrente máxima é de 50 mA;
- **GND (*ground*):** pino terra.

B) Entrada e Saída

Conforme mostrado na figura 1 (Legenda de Pinos do Arduíno Uno Rev3), a placa Arduíno Uno possui 14 pinos que podem ser usados como entrada ou saída digitais (de 0 a 13). Estes pinos operam em 5 V, onde cada pino pode fornecer ou receber uma corrente máxima de 40 mA. Também, convém dizer que cada pino possui resistência de pull-up interno que pode ser habilitado pelo *software*. Alguns desses pinos possuem funções especializadas:

- **PWM:** Pinos 3, 5, 6, 9, 10 e 11 podem ser usados como saídas PWM de 8 bits.
- **Comunicação serial:** Pinos 0 (RX) e 1 (TX) podem ser utilizados para comunicação *serial*. Deve-se observar que estes pinos são ligados ao microcontrolador responsável pela comunicação USB com o PC;
- **Interrupção externa:** Pinos 2 e 3 podem ser configurados para gerar uma interrupção externa;
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI, que embora compatível com o *hardware*, não está incluída na linguagem do Arduíno;
- **LED:** há um LED já montado e conectado ao pino digital 13.

Para interface com o mundo analógico, a placa Arduíno Uno possui 6 entradas, onde cada uma tem a resolução de 10 bits. Por defeito a referência do conversor AD (*analog-to-digital converter*) está ligada internamente a 5 V, ou seja, quando a entrada estiver com +5 V o valor da conversão analógica digital será 1023. O valor da referência pode ser mudado através do pino AREF.

C) Especificações Técnicas do Arduíno Uno

- Dimensão: 5,3 cm x 6,8 cm x 1,0 cm
- Microcontrolador: ATmega328P
- Tensão operacional: 5V
- Tensão de Alimentação (recomendada): 7-12V
- Tensão de Alimentação (limite): 6-20V
- Pinos de I / O digitais: 14 (das quais 6 são de saída PWM)

- Pinos de entrada analógicos: 6
- Corrente DC por pino I/O: 20 mA
- Corrente DC para Pin 3.3V: 50 mA
- Memória *flash*: 32 KB (ATmega328P), dos quais 0,5 KB usado pelo bootloader
- RAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Frequência de Clock: 16 MHz
- Temperatura de operação: 10º a 60 ºC

3.1.2 Arduíno IDE

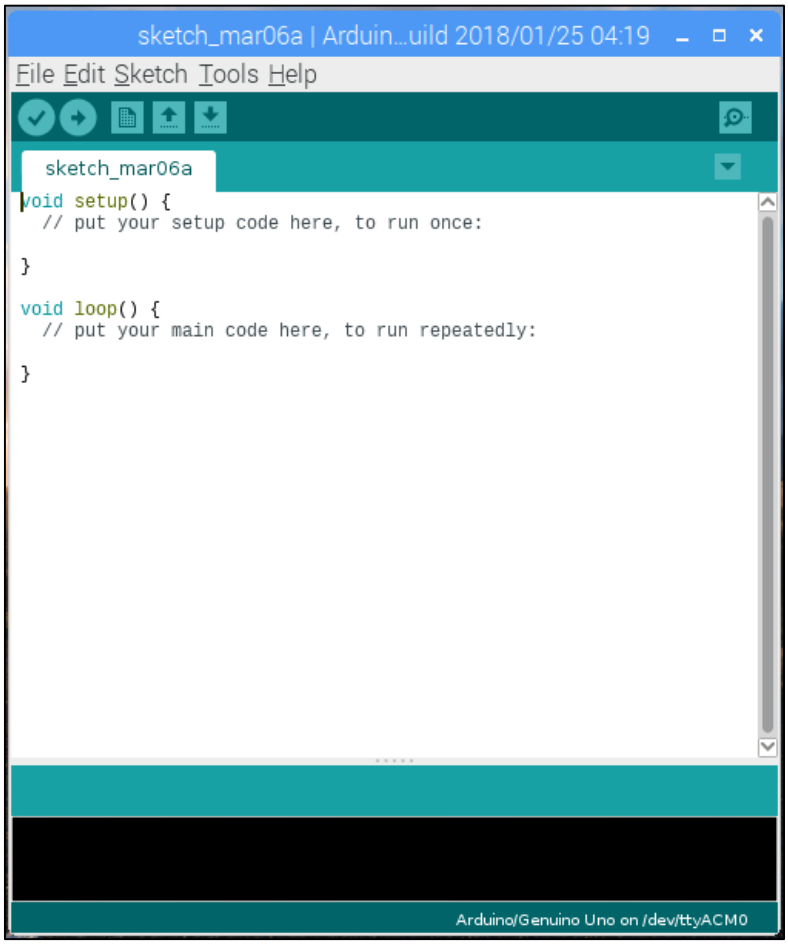
O *software* para programação do Arduíno é uma IDE (ambiente de desenvolvimento), que permite executar códigos num computador, conhecida como *sketch*, na qual é feita *upload* (envio de dados de um computador) para a placa do Arduíno Uno, através de uma comunicação *serial*.

Quando pressionado o botão *upload* da IDE, o código escrito é transmitido para o compilador avr-gcc, que realiza a tradução dos comandos para uma linguagem que pode ser compreendida pelo microcontrolador. A IDE possui uma linguagem própria baseada na linguagem C e C++.

O *sketch* feito pelo projetista, dirá à placa o que deve ser executado durante o seu funcionamento. Através da IDE é possível escrever programas que fazem interface com vários tipos de dispositivos como sensores, atuadores, visores, outros microcontroladores, além de possibilitarem uma forma rápida de desenvolvimento de protótipos.

Neste projeto será utilizado o Software Arduíno na versão 1.8.5, em que o *download* pode ser feita de forma gratuita no seguinte *site*: http://www.arduino.cc/en/Main/Software.

Figura 2: Arduíno IDE 1.8.5 no Raspberry Pi



Fonte: Elaboração própria

3.2 Estudo dos Microcomputadores

Além dos microcontroladores, existem outros sistemas que funcionam através do uso de microprocessadores. Desses sistemas, os mais encontrados são os microcomputadores.

Um microcomputador é um computador, caracterizado pela presença de um único microprocessador. Ele tem dimensões e capacidade computacional limitado. O microcomputador foi projetado para satisfazer as necessidades de um único utilizador cujo custo é bastante baixo.

O que difere o microcomputador do microcontrolador, é que no microcomputador a CPU, as memorias (voláteis e não voláteis) e as interfaces de seus periféricos (como teclado, monitor,

mouse, etc.) não estão presentes num único CI. Os microcomputadores realizam diversas funções ao mesmo tempo, muitas das quais provém da interação com os usuários, como, editar textos, executar jogos, gerar interfaces gráficas, além de outras funções, que se diferenciam daquelas normalmente realizadas por microcontroladores.

Para o projeto será utilizada o microcomputador Raspberry Pi 3 modelo B. O Raspberry Pi propõe-se a ser um computador completo, de baixo custo.

3.2.1 Raspberry Pi 3 Modelo B

Conhecido como um dos computadores mais baratos do mundo, Raspberry Pi ou simplesmente RPI, foi lançado para o mercado do consumidor no fim de Fevereiro de 2012 pela Raspberry Pi Foundation, com a proposta de ser um equipamento barato e fazer parte do currículo escolar, a fim de participar do processo educacional não só das crianças como também de estudantes ou de outras pessoas quaisquer.

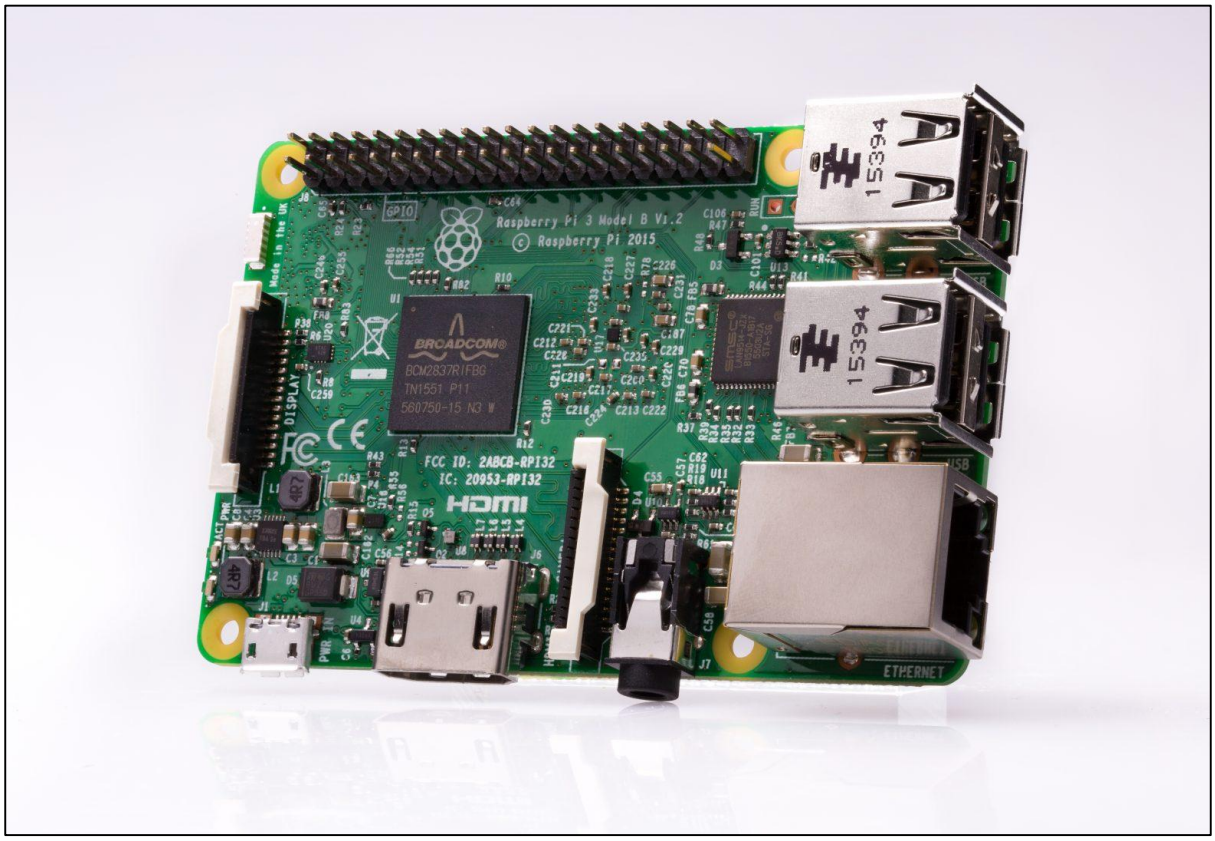
O Raspberry Pi é um computador completo, com considerável poder de processamento, numa placa de circuito impresso. A principal diferença do Raspberry Pi dos demais computadores está na dimensão deste equipamento, no qual assemelha-se a de um cartão de crédito.

É classificada como SBC (*Single-Board Computer*) ou seja, um computador montado em uma única placa com processador, memória, entrada/saída e outros componentes necessários para o seu funcionamento.

O Raspberry Pi é projetado para executar um sistema operativo chamado GNU/Linux ou simplesmente Linux. Diferente do Windows ou do OS X, o Linux é aberto, ou seja, é possível baixar o código fonte de todo o sistema operacional e fazer quaisquer alterações necessárias. Os *softwares* desenvolvidos para Windows ou OS X não funcionam nativamente no Linux. É de referir também que o Linux não é um exclusivo do Raspberry Pi.

O Raspberry Pi 3 Modelo B, lançado em 29 de Fevereiro de 2016, é o último modelo lançado até o momento da realização do projeto e, é com este modelo que o projeto será desenvolvido.

Figura 3: Raspberry Pi 3 Modelo B



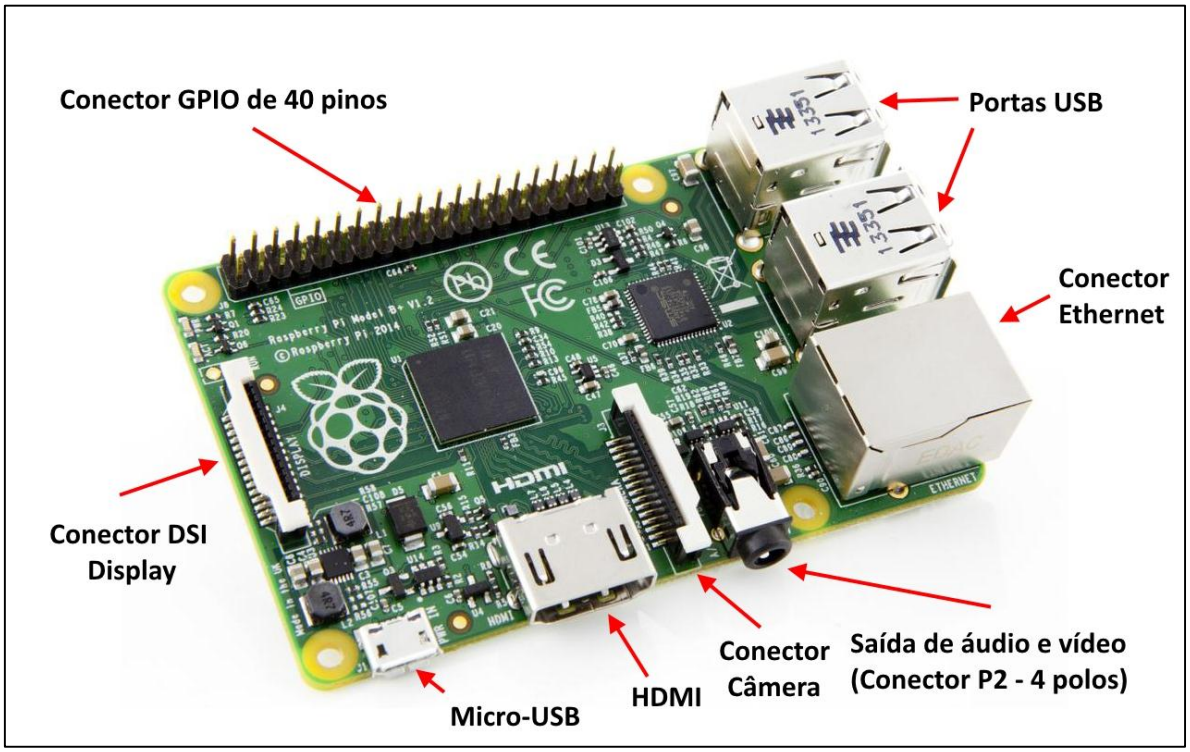
Fonte: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

O Raspberry Pi 3 Modelo B contém um processador Quad Core 1.2 GHz Broadcom BCM2837 de 64bit, 1 GB de RAM, 4 portas USB 2.0 com saídas de até 1.2A, conector de expansão de 40 pinos, saída de áudio/vídeo, interface para câmara, interface para display, entrada para cartão micro SD entre outras interfaces.

Possui também já integrados ao seu *hardware*, um adaptador Wi-Fi e um interface bluetooth 4.1, dessa forma beneficia o utilizador a não efetuar a compra de adaptadores externos, possibilitando assim a liberação de portas USB para interface com outros dispositivos.

O RPI 3B é capaz de desenvolver diversas aplicações tais como um computador convencional, por exemplo reprodução de vídeos em alta definição, folhas de cálculos, reprodução de jogos e entre outros.

Figura 4: Legenda de Pinos do Raspberry Pi 3 Modelo B



Fonte: https://www.filipeflop.com/blog/tutorial-raspberry-pi-linux/

A) Especificações Técnicas do Raspberry Pi 3 Modelo B:

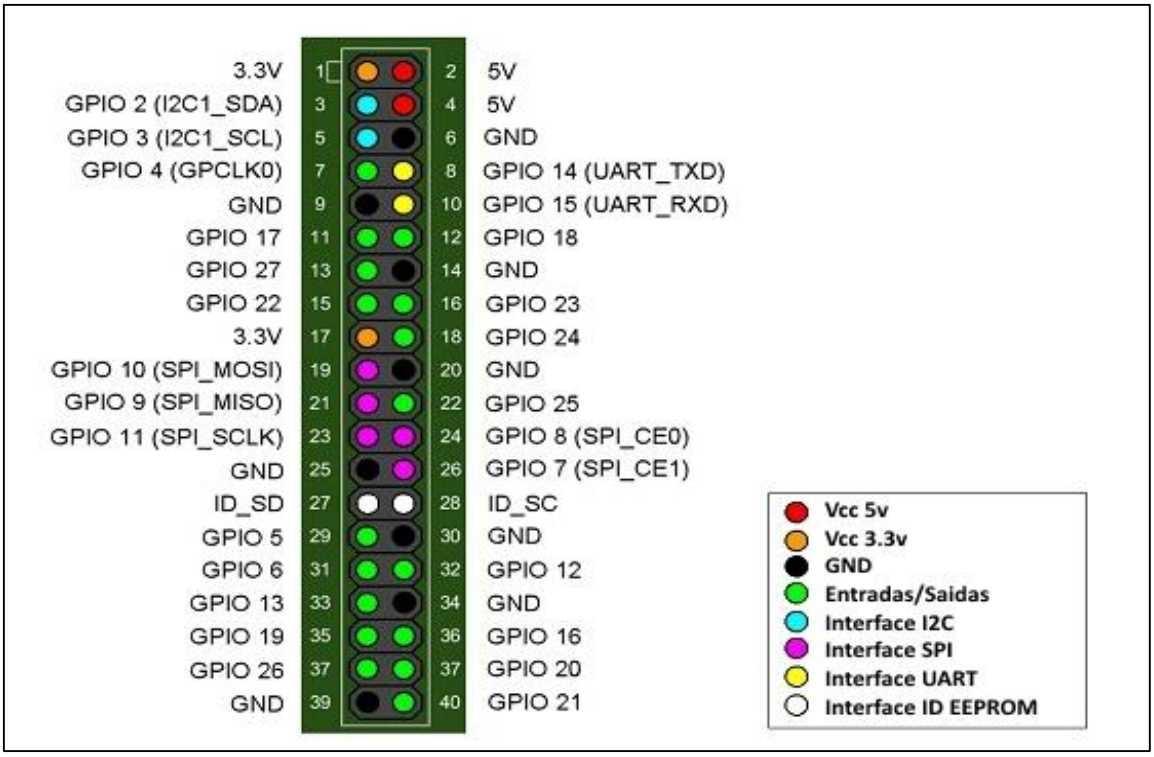
- Processador: Broadcom BCM2837 64bit ARMv8 Cortex-A53 Quad-Core;
- Frequência de Clock 1.2 GHz;
- Memória RAM: 1GB;
- Adaptador Wi-Fi 802.11n;
- Bluetooth 4.1 BLE (Bluetooth Low Energy);
- Conector de vídeo HDMI;
- 4 Portas USB 2.0;
- Conector Ethernet;
- Interface para câmara (CSI);
- Interface para display (DSI);
- Entrada para cartão microSD;
- Conector de áudio e vídeo;
- GPIO estendido de 40 pinos;
- Fonte de alimentação micro USB 5V 2.5A;
- Dimensões: 85 x 56 x 17mm.

B) Pinos de Entrada e Saída de Uso Geral (GPIO)

O GPIO é basicamente um conjunto de pinos, responsável por fazer a comunicação de entrada e saída de sinais digitais. No caso do RPI 3B, é composto por 40 pinos.

Esses pinos podem ser acessados para controlar *hardware* tais como LED's, motores, relés, etc., sendo esses como exemplos para saídas. Como entrada podem efetuar leitura de botões, interruptores, ou também leitura de sensores como, de temperatura, de luz, de movimento e entre muitos outros.

Figura 5: Diagrama de Pinos do GPIO de Raspberry PI 3 Modelo B



Fonte: https://www.filipeflop.com/blog/tutorial-raspberry-pi-linux/

Para entender o diagrama de uma forma mais clara, apresenta-se em baixo as características de cada uma:

- **Vcc 5V**: saída para alimentação de tensão de 5V. Deve-se ter atenção com o mesmo, pois não pode entrar em contato com as outras portas, por forma a evitar curto-circuitos entre os pinos.

- **Vcc 3.3V**: saída para alimentação, porém com uma tensão de 3.3V. As portas do RPI funcionam no nível 3.3V. É possível ligá-los a portas com nível de 5V recorrendo a dispositivos de conversão de nível.
- **GND**: pino de terra (*ground*).
- **Entrada e Saída**: servem para enviar e receber dados digitais.
- **Interface I2C**: essas portas podem ser programadas para interface I2C (fazem conexões entre periféricos de baixa velocidade). Utiliza-se um barramento entre dois fios, sendo um de dados e outro de clock (frequência), para comunicação *serial* entre circuitos integrados montados em uma mesma placa.
- **Interface UART**: essas são as portas *serial*, que utilizam o protocolo RS-232 para enviar e receber sinal digital. Não existe um sinal de *clock* para sincronizar o sinal transmitido com o sinal recebido. O UART é um protocolo de comunicação *full-duplex* (pode receber e enviar dados ao mesmo tempo), ou seja, a comunicação é realizada ao mesmo tempo de um dispositivo para o outro, pois os dados transitam por meios diferentes.
- **Interface SPI**: esses pinos são também para entrada e saída de dados digitais. Porém, com esses pinos é possível fazer uma comunicação *serial full-duplex* síncrono, que permite o processador do Raspberry Pi comunicar com algum periférico externo de forma bidirecional. Mas essa comunicação só acontece, se e somente se o protocolo for implementado.
- **Interface ID EEPROM**: são as portas do ID EEPROM, um tipo de memória que pode ser programado e apagado várias vezes, através de uma tensão elétrica interna ou externa.

C) Alimentação

O RPI 3B é alimentado pelo pequeno conector micro USB encontrado na lateral esquerda inferior da placa. Esse conector é o mesmo que encontramos na maioria dos *smartphones*, tabletes e/ou outros dispositivos.

Alguns carregadores destinados para smartphones podem funcionar com o Raspberry Pi. O RPI 3B consome mais energia do que a maioria dos dispositivos micro USB e requer uma fonte de

tensão de 5V 2500 mA para poder funcionar. Contudo, deve-se ter o cuidado com alguns carregadores, alguns podem fornecer energia insuficiente, causando problemas intermitentes na operação do RPI.

Conectar o RPI à uma porta USB de um computador é possível, mas não é recomendado, uma vez que, assim como os carregadores de menor porte, as portas USB de um computador não conseguem fornecer a potência requerida para que o RPI trabalhe corretamente.

Por não possuir um botão de "ligar" e "desligar", deve-se conectar a fonte de energia micro-USB apenas quando estiver pronto para começar a usar. Sendo assim começará a funcionar no instante em que for conectado à energia e apenas poderá ser desligado se removermos, fisicamente, o cabo de energia. Todavia deve-se encerrar a execução do sistema operativo antes de se desligar o RPI da alimentação.

D) Sistemas Operacionais Suportados

O RPI 3B é compatível com sistemas operacionais baseados em Linux, por exemplo Debian, Arch Linux, Ubuntu (Ubuntu Mate e Snappy Ubuntu Core), Raspbian (baseado no Desbian, é a variante linux oficial do RPI) Windows 10 IoT Core, OSMC entre outros. São todos oficialmente suportados pelo RPI.

No presente projeto, optar-se-á pela instalação do sistema operacional Raspbian no RPI, visto que é a variante Linux oficial do RPI.

E) Entrada de cartão micro SD

O RPI não possui o disco rígido real como um computador convencional, logo, o cartão micro SD é usado como SSD (*solid state drive*), que é destinado à instalação do sistema operacional e todos os outros *softwares*, bem como na armazenagem de arquivos.

Para o RPI 3B será necessário um cartão micro SD de pelo menos 8GB, e deve ser um cartão de classe 10 (o número de classe de um cartão SD determina a velocidade mínima de gravação sequencial), uma vez que estes cartões são capazes de transferir pelo menos 10MB/seg.

A entrada do cartão micro SD fica localizada na parte posterior da placa.

F) Saída HDMI

Esta porta de saída é usada para conectar o Raspberry Pi com um monitor via HDMI (*High Definition Multimedia Interface*). Assim, qualquer monitor ou TV que possui entrada HDMI pode ser conectada ao RPI. No entanto, se o monitor possuir entrada VGA (*Video Graphics Array*), deve-se utilizar um conversor HDMI-VGA.

G) Entrada Ethernet e USB

Tanto a porta Ethernet como a porta USB do RPI 3B são fornecidos através do *chip* LAN9514 integrado. Contem USB 2.0 de alta velocidade com um controlador Ethernet 10/100. As portas USB são usadas para conectar as entradas (teclado, *mouse*, etc.). Quase tudo que pode ser conectado ao computador convencional via USB também pode ser conectado ao Raspberry Pi.

H) Saída de Vídeo RCA e Saída de Áudio

O conetor de áudio e RCA está presente na placa e destina-se saída de áudio e vídeo. Mesmo sabendo que o RPI suporta o som em sua saída HDMI, há uma entrada de áudio padrão de 3,5 mm para conectar auriculares. Para o vídeo, a entrada RCA envia vídeo para qualquer dispositivo de vídeo RCA conectado.

3.3 Estudo dos Sensores Utilizados

Segundo definições dadas por alguns autores, os sensores são:

Definição 1: *"Sensores são dispositivos usados para detetar, medir ou gravar fenômenos físicos tais como calor, radiação etc., e que respondem transmitindo informação, iniciando mudanças ou operando controles."* [Moreira, 2002]

Definição 2: *"São dispositivos eletreletrónicos que tem a propriedade de transformar em sinal elétrico a transformação de uma grandeza física que está relacionada a uma ou mais propriedades do material de que é feito o sensor."* [Autor Desconhecido, extraído em: http://www.if.ufrgs.br/mpef/mef004/20061/Cesar/SENSORES-Definicao.html]

Definição 3: *"Sensores são dispositivos que variam suas propriedades sob a ação de uma grandeza física, podendo fornecer, direta ou indiretamente, um sinal que indica essa grandeza."* [BASTOS, 2002]

Os sensores e os sistemas de obtenção de dados possuem características importantes, tais como sensibilidade, precisão, faixa de atuação, estabilidade, tempo de resposta, histerese, linearidade e outros.

Segundo BASTOS (2002):

- **Sensibilidade –** é a resposta de um sensor em relação à variação do estímulo para permitir um sinal de saída detetável.
- **Precisão –** é o grau de liberdade dos erros aleatórios ou o grau de repetibilidade e de reprodutibilidade.
- **Intervalo de indicação ou atuação –** é o conjunto de valor limitado pelas indicações extremas ou valores máximos e mínimos que podem ser medidos pelo sensor.
- **Estabilidade –** é a não variação de suas características físicas em relação ao tempo de uso (é uma característica desejável do sensor).
- **Tempo de resposta –** é o tempo que um sensor precisa para transformar a variação de um estímulo em sinal elétrico.
- **Histerese –** é a característica de um sensor retornar ao seu estado físico inicial com a variação do estímulo.
- **Linearidade –** é uma característica desejável de um instrumento, onde a leitura deste é linearmente proporcional à grandeza a ser medida.

Os principais pontos que foram avaliados para a escolha dos sensores são:

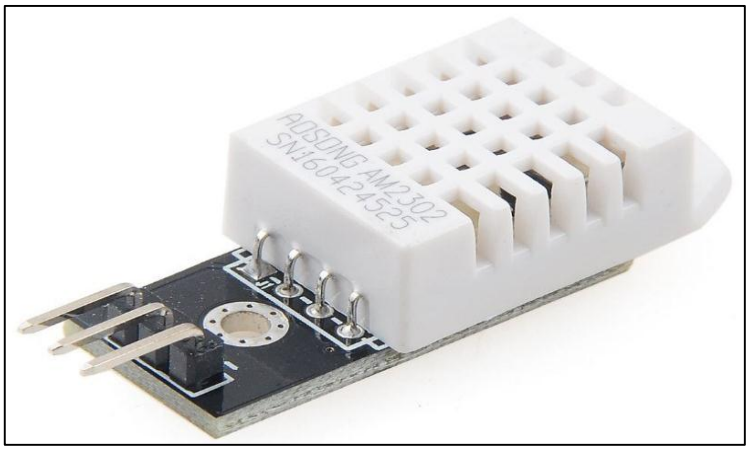
- Faixa de operação;
- Precisão;
- Resolução ou Sensibilidade;

- Corrente de funcionamento;
- Custo e disponibilidade no mercado exterior.

3.3.1 Sensor DHT22

Para medição das variáveis humidade e temperatura, foi escolhido o sensor DHT22. O DHT22 é um sensor de temperatura e humidade que permite fazer leituras de temperaturas entre -40 a +80 graus Celsius e humidade entre 0 a 100%, sendo assim de fácil uso com Arduíno, Raspberry Pi e outros microcontroladores, pois possui apenas 1 pino com saída digital.

Figura 6: Sensor DHT22



Fonte: https://www.amazon.com

A) Especificações Técnicas

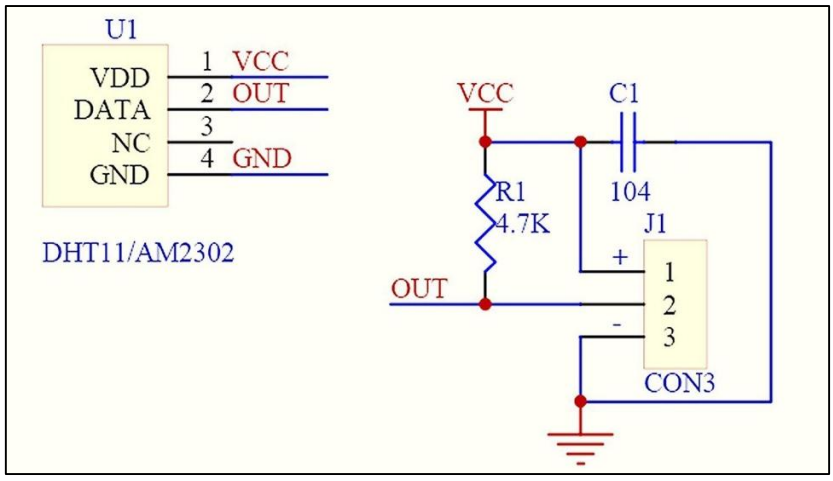
- Tensão de operação: 3-5V (5,5V);
- Faixa de medição de humidade: 0 a 100% UR;
- Faixa de medição de temperatura: -40º a +80ºC;
- Corrente: 2,5mA máxima durante uso, em *standby* de 100uA a 150 uA;
- Precisão de medição de humidade: ± 2,0%;
- Precisão de medição de temperatura: ± 0,5 ºC;
- Resolução: 0,1;
- Tempo de resposta: 2 seg.;
- Dimensões: 25 x 15 x 7mm (sem terminais).

B) Princípio de Funcionamento

O princípio de funcionamento do DHT22 é do tipo capacitivo. Um sensor capacitivo funciona basicamente através de um condensador que exibe uma variação do valor nominal da sua capacitância (ou capacidade elétrica) em função da variação de uma grandeza não elétrica. Uma vez que um condensador consiste basicamente num conjunto de duas placas condutoras, separadas por um dielétrico, as variações no valor nominal da capacidade podem ser provocadas por redução da área frontal, da separação entre as placas, ou por variação da constante dielétrica do material. Os sensores capacitivos permitem medir, com grande precisão, um grande número de grandezas físicas, tal como a humidade relativa do ar.

O funcionamento de um sensor capacitivo de humidade (designado sensor higrométrico) explora a dependência da constante dielétrica de alguns materiais com o teor de água no ar ambiente. O dielétrico é, neste caso, constituído por uma película fina de um material simultaneamente isolador e higroscópico o qual, dada a natureza porosa de um dos dielétricos, encontra-se em contato com o ambiente cuja humidade relativa pretende-se medir.

Figura 7: Esquema Elétrico do DHT22



Fonte: https://www.amazon.com

Tabela 1: Pinos do DHT22 (da esquerda para direita)

Pinos	Nome	Função
1	Vcc	Alimentação
2	Dados	Transmissão de Dados
3	-	Não Usado
4	GND	Terra

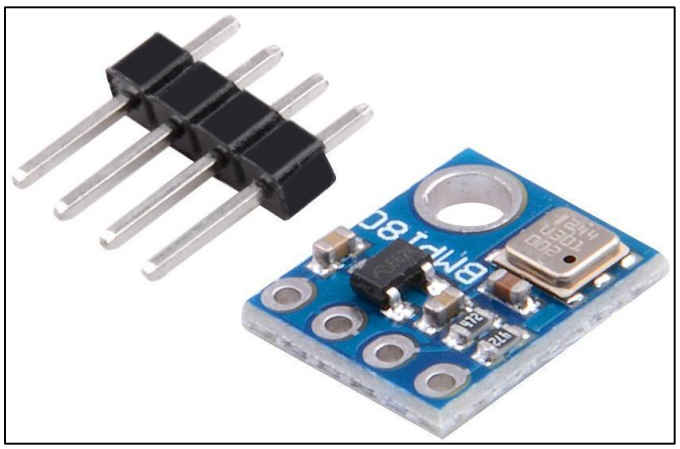
Fonte: Elaboração própria

3.3.2 Sensor BMP180

Para medição das variáveis pressão e altitude, foi escolhido o sensor BMP180. O BMP180 é um sensor que mede a pressão atmosférica, a temperatura e a altitude. Contudo, neste projeto será utilizada somente para medição da pressão atmosférica e altitude visto que já há um sensor para efetuar a medição da temperatura.

É um Sensor que pode ser otimizado para uso em telemóveis, PDAs (*Personal digital assistants*), dispositivos de navegação GPS e equipamentos externos. Com um ruído de baixa altitude (apenas 0,25m em tempo de conversão rápido) o BMP180 oferece um bom desempenho. A interface I2C permite a fácil integração do sistema com microcontroladores. Possui alta precisão e linearidade, além de estabilidade a longo prazo.

Figura 8: Sensor BMP180



Fonte: https://www.amazon.com

A) Especificações Técnicas

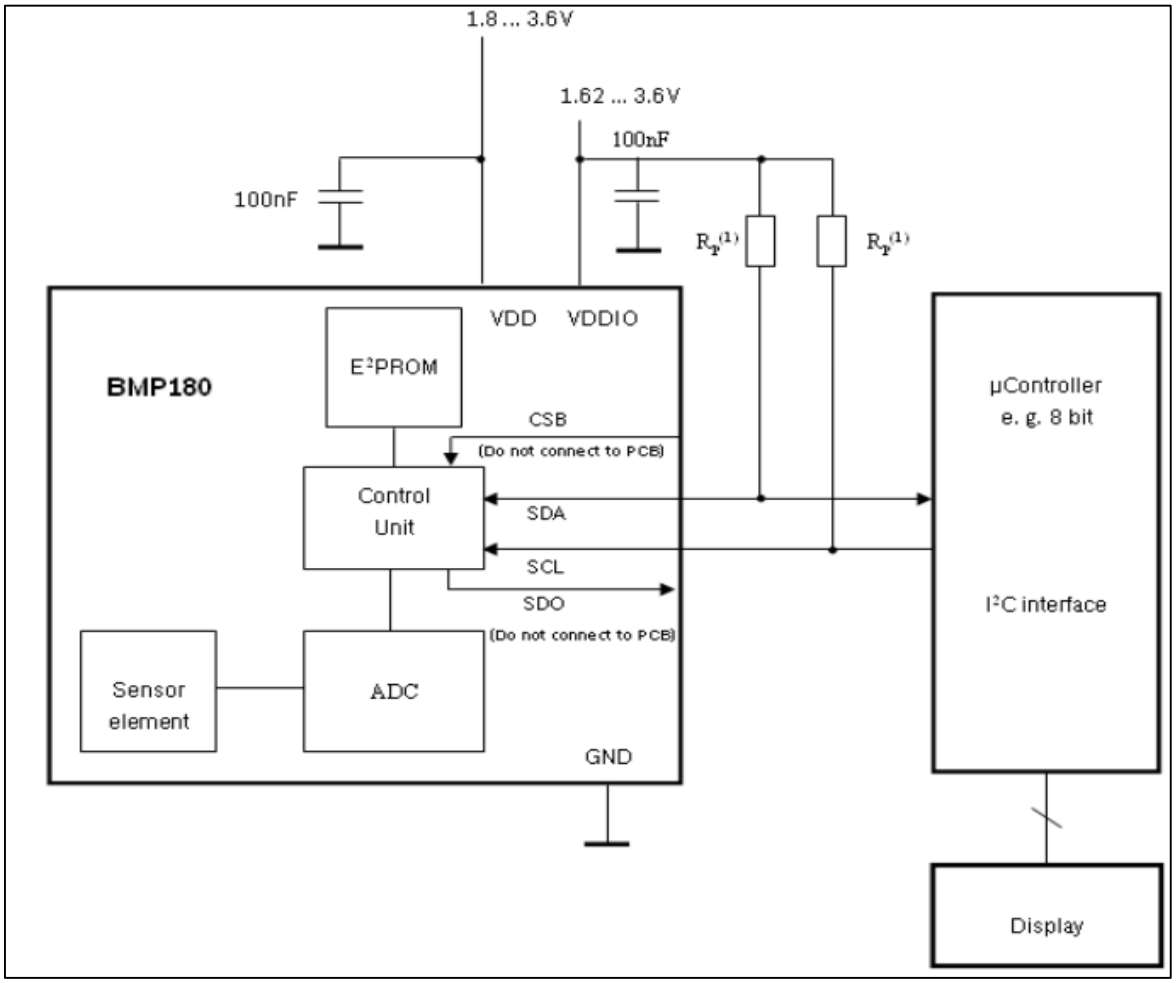
- Tensão de operação: 1,8 a 3,6V;
- Faixa de medição para pressão atmosférica: 300 a 1100hPa;
- Altitude: -0,5km a +9km;
- Temperatura de operação: -40° a 85°C (precisão total 0° a 85°C);
- Faixa de medição de temperatura: 0° a 65°C;
- Resolução: 0,06hPa (Alt. 50cm);
- Corrente de medição: 5 μA;
- Corrente de espera: 0,1 μA;
- Precisão do sensor barométrico: ± 0,12 hPa;
- Tempo de reação: 7,5ms;
- Dimensões (CxLxE): 13x10x2mm.

B) Princípio de Funcionamento

O princípio de funcionamento do BMP180 é do tipo piezo-resistivo. Um material piezelétrico é aquele que, quando exercido uma força sobre o material este traduz a força em diferença de potencial e vice-versa. Um sensor piezo-resistivo funciona da mesma forma, ou seja, inverte a força exercida sobre ele em uma diferença de resistência.

Quando ocorrem variações da pressão, ocorrem igualmente alterações nas resistências implantadas, de acordo com o efeito piezo-resistivo. Esta resistência é alimentada por uma carga, parte de um sistema de controlo o qual pode ser analisado. A medição é obtida a partir deste sistema de controlo, o qual traduz para uma saída de 4 a 20 mV que é convertida em um sinal digital por um conversor Analógico/Digital interno e, posteriormente, enviada para o microcontrolador.

Figura 9: Circuito de Aplicação Típica



Fonte: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf

Tabela 2: Pinos do BMP180 (da esquerda para direita)

Pinos	Nome	Função
1	Vcc	Alimentação
2	GND	Terra
3	SCL	Entrada de Clock Serial
4	SDA	Entrada de Dados Serial

Fonte: Elaboração própria

3.3.3 Sensor SEN0170

Para medição da variável velocidade de vento, foi escolhido o sensor SEN0170. O SEN0170 é um instrumento que realiza a medição da velocidade do vento, ou seja, trata-se de um anemômetro, composto por três copos de vento e módulo de circuito.

Esse anemômetro é feito de liga de alumínio que utiliza a tecnologia especial de moldagem de precisão do molde, com uma tolerância muito pequena, uma precisão de superfície bastante alta, uma alta resistência às tempestades e à corrosão, e é à prova de água.

O cabo de conexão é anticorrosivo, evitando assim o desgaste derivado a erosão, garantindo assim o uso deste instrumento por um longo tempo.

Este produto pode ser perfeitamente utilizado em máquinas de engenharia (guindaste, guindaste de esteira, guindaste de porta, guindaste de torre, etc.), ferrovias, portos, centrais de energia eólica, meteorologia, teleférico, meio ambiente, estufa, reprodução, ar condicionado, monitoramento de energia, agricultura, saúde, etc.

Através de um microcontrolador, com as instruções e os códigos adequados, os usuários podem facilmente obter o valor da velocidade do vento através da saída do sinal de tensão (0 a 5V).

Figura 10: Anemômetro SEN0170

Fonte: https://www.amazon.com

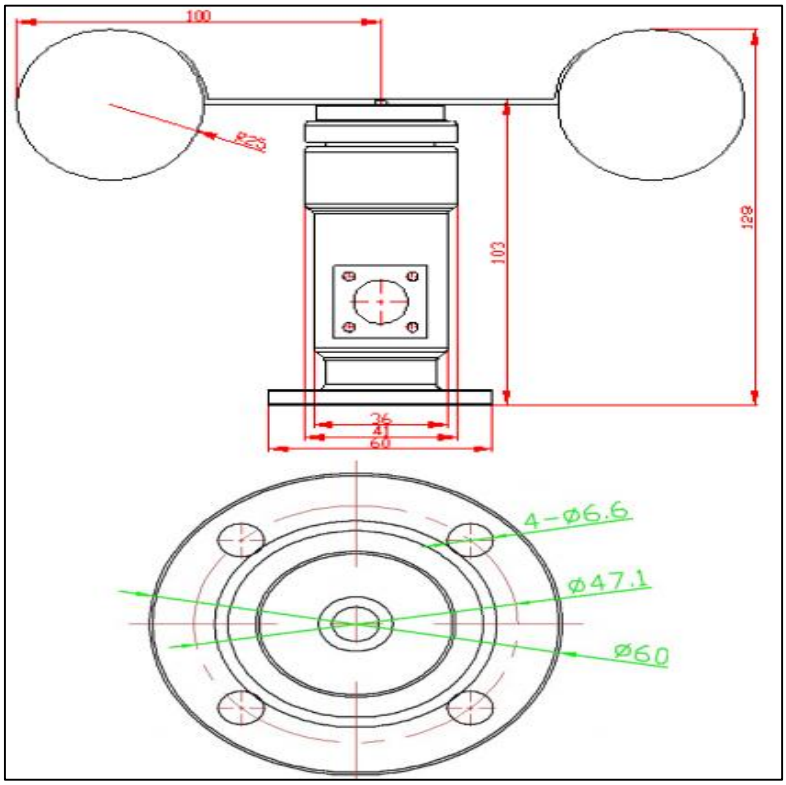
A) Especificações Técnicas

- Material: liga de alumínio;
- Modo de sinal de saída: 0-5V (sinal de tensão);
- Tensão de alimentação: DC 9-24V;
- Consumo de energia: Voltagem Máxima ≤ 0.3W;
- Início da velocidade do vento: 0.4-0.8m/s;
- Resolução: 0.1m/s;
- Intervalo efetiva de medição da velocidade do vento: 0-30m/s;
- Erro do sistema: ± 3%;
- Distância de transmissão: mais de 1000m;
- Modo de transmissão: por cabo;
- Modo de conexão: sistema de três fios;
- Temperatura de operação: -40 ℃ ~ 80℃;
- Humidade de operação: 35% ~ 85%;

B) Características:

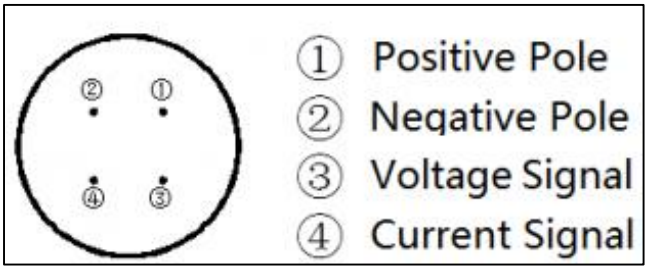
- Alta dureza;
- Proteção contra corrosão;
- Impermeabilidade;
- Alta precisão.

Figura 11: Diagrama de dimensão do SEN0170



Fonte: https://www.dfrobot.com

Figura 12: Pinos do SEN0170



Fonte: https://www.dfrobot.com

Tabela 3: Designação dos pinos do SEN0170

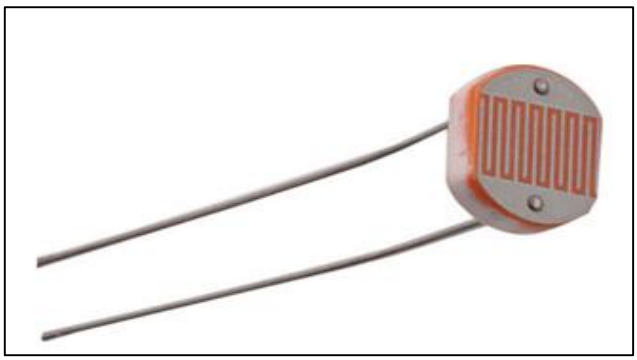
Pinos	Nome	Função	Cor de Fio
1	Vcc	Alimentação (9-24V)	Vermelho
2	GND	Terra	Preto
3	Voltage Signal	Sinal de tensão	Amarelo
4	Current Signal	Sinal de corrente	Azul

Fonte: Elaboração própria

3.3.3 LDR

Para medição da variável intensidade luminosa, foi escolhido o sensor LDR (*Light Dependent Resistor*). O LDR é um tipo de resistência especial que apresenta uma mudança na sua característica de resistência elétrica quando submetido à ação da luz.

Figura 13: LDR



Fonte: http://www.ca.diigiit.com/ldr

A) Especificações Técnicas

- Diâmetro: 5mm;
- Comprimento com terminal: 34mm;
- Peso: 0,4g;
- Resistência à luz: ~ 1k Ohm;
- Resistência à escura: ~ 10k Ohm;
- Tensão máxima: 150V;
- Potência máxima: 100mW.

B) Princípio de Funcionamento

O LDR é fabricado com materiais de alta resistência, como por exemplo o Sulfeto de Cádmio (CdS) ou o Sulfeto de Chumbo (PbS). Esses materiais possuem poucos eletrões livres quando colocados num ambiente escuro e libertam eletrões quando há incidência de luz sobre eles, aumentando sua condutividade. A esse efeito dá-se o nome de Fotocondutividade.

Quando a incidência de luz cessar sobre o componente, os eletrões retornam à camada de valência e a resistência do material volta a aumentar.

Nota: É importante destacar que o LDR será utilizado no projeto, juntamente com uma resistência de10k Ohm, para simular um sensor TSL2561 que mede a variável luminosidade, pois o LDR é uma resistência especial que varia de valor consoante a luz que inside nele.

IV. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

4.1 Linguagem em C++

Segundo BALTAREJO, Paulo e SANTOS, Jorge (2006) C++ é *"uma linguagem de programação genérica que suporta tanto os paradigmas da programação estruturada como o orientado ao objeto."* A partir de 1990, o C++ tornou-se uma das linguagens de programação mais populares. Bjarne Stroustrup desenvolveu o C++ (originalmente designado "C com classes") nos laboratórios da Bell em 1983, como um melhoramento da linguagem C. Estes melhoramentos incluíram: adição de classes, funções virtuais, sobrecarga de operadores, múltipla herança, templates e tratamento de exceções.

Principais Características:

- **Programação Orientada à Objetos:** a possibilidade de utilizar programação orientada à objetos permite ao programador projetar aplicações de um ponto de vista mais parecido com comunicação entre objetos que de uma sequência estruturada de código. Além disso, permite a reusabilidade de código (eliminar a repetição de código) de uma forma mais lógica e produtiva. A linguagem foi desenvolvida com o cuidado de prover atríbutos Orientados à Objeto para a linguagem C sem comprometer a eficiência;
- **Portabilidade:** pode-se praticamente compilar o mesmo código C++ em qualquer tipo de computador e sistema operacional sem fazer grandes mudanças. C++ é uma das mais usadas e portadas linguagens de programação;
- **Brevidade:** código escrito em C++ é muito menor em comparação com outras linguagens, desde o uso de caracteres especiais e preferidos antes de palavras-chave, evitando esforço;
- **Programação Modular:** um corpo de aplicação em C++ pode ser feita de vários arquivos de código que serão compilados separadamente. Economizando tempo, pois não é necessário recompilar toda a aplicação quando se faz uma mudança simples, mas apenas aquele arquivo que a contem. Esta característica também permite C++ ligar com código produzido em outras linguagens como o Assembly ou C;
- **Compatibilidade com C:** qualquer código escrito em C pode ser facilmente incluído em um programa C++ sem fazer grandes mudanças;

- **Velocidade:** o código resultante de uma compilação C++ é muito eficiente, devido a sua dualidade de linguagem de Alto e Baixo nível e do tamanho reduzido da linguagem em si;
- Não há um "dono" da linguagem C++. Há vários compiladores e sistemas operacionais que utilizam o padrão C/C++ ANSI. Isto significa na prática que esta linguagem tem inúmeros patrocinadores, famosos e anônimos. Isto mantém o suporte sempre muito atualizado e disponível pela internet.
- C++ é desenvolvido para fornecer ao programador múltiplas escolhas, mesmo que seja possível ao programador escolher a opção errada.

A razão pela escolha dessa linguagem deve-se ao fato do Arduíno IDE trabalhar com a linguagem em C++, onde será programada o código para realização da leitura dos valores dos respetivos sensores.

4.2 Linguagem Python

Segundo NITERÓI (2009), Python é *"uma linguagem de programação interpretada, de código-fonte aberto e disponível para vários sistemas operacionais."* Ser uma linguagem interpretada significa dizer que ao escrever-se um programa, este não será compilado (traduzido para uma linguagem de máquina), mas sim lido por um outro programa (chamado de interpretador) que traduzirá para a máquina o que seu programa quer dizer.

O interpretador para Python é interativo, ou seja, é possível executá-lo sem fornecer um programa para ele. Ao invés disso, o interpretador disponibilizará uma interface interativa onde é possível inserir os comandos desejados, um por um, e ver o efeito de cada um deles.

As principais características são:

- Python é uma linguagem de programação que não é necessário compilar seu código para que a máquina entenda;
- Roda em ambientes Linux, Windows, MacOS, smartphones e outra infinidade de sistemas;

- Por padrão ela é uma linguagem totalmente orientada a objetos, ela permite que o programador desenvolva de forma funcional;
- É um *software* Livre, ou seja, é gratuita;
- Possui código aberto, logo não requer preocupações com a estabilidade da linguagem no mercado, já que possui uma imensa comunidade ao redor do mundo;
- O Python é uma linguagem multiuso, pois permite criar desde aplicações *desktop* a *websites*.

A sua escolha deve-se ao fato de o Raspberry Pi trabalhar em ambiente Linux, e com esses tipos de sistemas operacionais a linguagem de programação Python é o mais recomendado, pelo fato das biliotecas existentes em Python que permitem-nos aceder aos pinos GPIO estão muito desenvolvidas e livremente disponíveis devido à grande comunidade de desenvolvedores existente.

A linguagem Python será utilizada no desenvolvimento do código para obtenção dos dados *serial* vindos do Arduíno e envia-los para o banco de dados MySQL.

4.3 Banco de Dados MySQL

Segundo ALECRIM (2008), o MySQL é *"um sistema de gerenciamento de banco de dados (SGBD), das mais populares que existe e, por ser otimizado para aplicações de internet, é amplamente utilizado na internet."* Utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como interface.

É muito comum encontrar serviços de hospedagem de *websites* que oferecem o MySQL e a linguagem PHP, justamente porque ambos trabalham muito bem em conjunto.

Outro fator que ajuda na popularidade do MySQL é sua disponibilidade para praticamente qualquer sistema operacional, como Linux, Windows e Mac OS X. Além disso, o MySQL é um *software* livre (sob licença GPL), o que significa que qualquer um pode estudá-lo ou alterá-lo conforme a necessidade.

Como principais características temos:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Alta compatibilidade com linguagens como PHP, Java, Python, C#, Ruby e C/C++.
- Baixa exigência de processamento (em comparação como outros SGBD);
- Vários sistemas de armazenamento de dados (database engine), como MyISAM, MySQL Cluster, CSV, Merge, InnoDB, entre outros;
- Recursos como transações, conectividade segura, indexação de campos de texto, replicação, etc.;
- Instruções em SQL, como indica o nome.

A sua escolha deve-se ao fato, não só por ter um prévio conhecimento deste tipo de SGBD, como também de ser um *software* livre e que permite uma fácil integração com a linguagem PHP. Ainda, encontra-se disponível juntamente com o servidor Apache num único pacote, através do WampServer para ambiente Windows ou XampServer para ambiente Linux permitindo uma administração e configuração do sistema centralizada numa única máquina.

4.4 Phpmyadmin

Phpmyadmin é o administrador de todo o servidor MySQL desenvolvido em PHP e HTML, ou seja, ele é uma aplicação. Tem a função de levar MySQL para um navegador, tornando mais fácil gerenciar o banco de dados. Phpmyadmin pode fazer várias coisas no MySQL, mas das principais e importantes estão listadas a seguir:

1) Navegar e eliminar bancos de dados, tabelas, visualizações, colunas, etc.;
2) Importação e exportação de banco de dados;
3) Criar, atualizar, eliminar, renomear e alterar bancos de dados, tabelas, colunas e outros mais;
4) Administração de usuários e privilégios do MySQL;
5) Criação e leitura de *Dumps* (contém um registo da estrutura da tabela e/ou dos dados de um banco de dados e geralmente está na forma de uma lista de instruções SQL) de tabelas;

6) Testar e criar *queries* (conjunto de instruções que permite extrair repetidamente determinado subconjunto de dados).

Os requisitos para instalação do Phpmyadmin estão listados a seguir:

1) Servidor WEB: para o projeto, Apache2;
2) PHP: para o projeto, PHP na versão 7.0;
3) Navegador da Internet;
4) Banco de Dados: compatível com o MySQL.

4.5 Linguagem PHP

Para VASCONCELOS (2013), "*PHP ("Hypertext Pre-processor", inicialmente "Personal Home Page") é uma linguagem interpretada livre e utilizada para gerar conteúdo dinâmico para internet. É uma linguagem de programação de domínio específico, ou seja, seu objetivo estende-se a um campo de atuação que é o desenvolvimento WEB, embora tenha variantes como o PHP-GTK (com foco em desenvolvimento Desktop)*".

Segundo CASTELA (2010), o "*PHP diferencia-se de outros scripts porque ao invés de escrever-se um monte de comandos para imprimir os HTML, é escrito um arquivo HTML com os códigos PHP embutidos entre o HTML delimitado por tags ("<" e ">") de inicio a fim*".

A vantagem do PHP consiste no fato de ser uma multiplataforma, podendo ser usado na maioria dos Sistemas Operacionais, fonte aberto, e diferente de *scripts*, como o JavaScript, ele é executado no servidor, que é suportado pela maioria dos servidores WEB que existem hoje no mercado, como o Apache, IIS, PWS, etc. O cliente recebe apenas os resultados dos *scripts*, que são interpretados no servidor, sem ter acesso ao código.

As principais características são:

- Velocidade e robustez;
- Estruturado e orientação a objetos;
- Portabilidade (independência de plataforma);

- Dinâmica;
- Sintaxe similar a C/C++ e o Perl;
- Fonte aberta;
- *Server-side* (o cliente manda o pedido e o servidor responde em página HTML).

A escolha desta linguagem deve-se ao facto desta linguagem ser altamente compatível com o MySQL e outras linguagens que serão necessários para o desenvolvimento do *website*.

4.6 Google Charts

Segundo o instrutor Luiz Carlos Diniz, do *website* School of Net, o Google Charts é uma ferramenta para construção de gráficos profissionais, tabelas, organogramas e outras coisas mais, através das linguagens HTML, JavaScript, entre outros. Com esta ferramenta, pode-se criar vários tipos de gráficos num *website*, além de exportar dados, integrações com planilhas no Google Drive e consultas dinâmicas aos dados do gráfico.

O Google Charts será utilizado para criar medidores em tempo real, e gráficos exibindo a variação de temperatura, humidade e velocidade de vento ao longo do tempo.

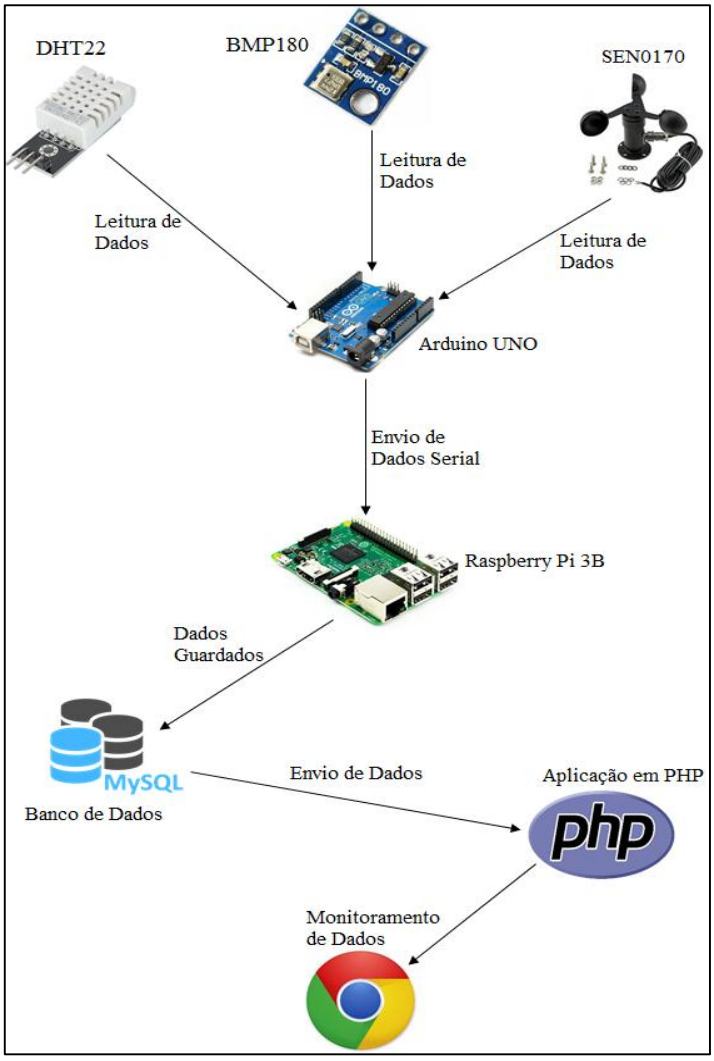
A razão da escolha deve-se ao fato de ser compatível com PHP e MySQL, sendo que as ferramentas do Google Charts são poderosas, de uso simples e gratuitas.

V. IMPLEMENTAÇÃO DO PROJETO

5.1 Descrição do Sistema

O Arduíno Uno faz a leitura dos sensores e envia os dados para um microcomputador Raspberry Pi. No RPI os dados são processados e armazenados num banco de dados (nesse caso o MySQL), ficando disponíveis posteriormente para consulta em tempo real, através de um aplicativo de monitoramento feito em PHP, juntamente com as ferramentas do Google Charts, como exemplifica o esquema abaixo. Estes dados poderão ser acessíveis através de um *website* simples por outros dispositivos tais como computadores, *smartphones*, etc.

Figura 14: Arquitetura Global do Sistema



Fonte: Elaboração Própria

5.2 Configuração do Raspberry Pi

5.2.1 Instalação do Sistema Operativo no RPI

O Raspberry Pi é um pequeno computador e por isso, deve ser instalado nele um sistema operativo (SO). E como o RPI não possui disco rígido, o sistema operativo é instalado na memória externa. Para isso, o cartão de memória (micro SD) é usado para a instalação do SO, assim como todos os outros *softwares* e arquivos necessários de suporte são armazenados no mesmo.

Dentro dos diferentes tipos de sistemas operacionais existentes, optou-se pelo NOOBS (*New Out of the Box Software*) que contém o Rasbian incluído, pois, é o mais recomendado para iniciantes.

A seguir, serão indicados, passo a passo, todos os processos, desde a formatação do cartão micro SD até a instalação do SO (Rasbian):

1) Ter um cartão micro SD de pelo menos 8GB e de preferência classe10;
2) Formatar o cartão micro SD usando o *software* SDFormatter como formato FAT;
    - O *download* do SDFormatter pode ser feita consoante o SO do computador. Pode ter acesso ao *link* fornecido e realizar o *download*: https://www.sdcard.org/downloads/formatter_4/;
    - Instalar no computador;
    - Inserir o cartão SD no leitor de cartão SD do computador;
    - Executar o *software* e escolher a unidade.
3) Fazer o *download* da versão mais recente do NOOBS (arquivo zip) no computador a partir do *link*: https://www.raspberrypi.org/downloads/;
4) Extrair o arquivo e copiar para o cartão micro SD formatado. O tamanho do arquivo é cerca de 1,47GB;
5) Conectar o teclado, o *mouse* e o monitor ao RPI, inserir o cartão micro SD e, finalmente, ligar a alimentação e a instalação do SO é iniciada;
6) No início da instalação, há uma listagem de SO em que deve-se escolher um deles. Recomenda-se escolher "Raspbian" e clicar no ícone "instalar". A instalação levará cerca de 30 minutos para completar;

7) Após a conclusão, a ferramenta de configuração do *software* Raspberry Pi é aberta onde pode-se configurar o sistema para usar do modo que desejar-se. Existem algumas configurações úteis necessárias para serem feitas. Por exemplo, clicar em *"Advance Option"* e habilitar SSH, para o fuso horário e configuração do idioma, clicar na opção *"Internationalization"*;
8) Na conclusão da instalação, recebe-se uma mensagem como "*OS (es) installed successfully*" e o ambiente de trabalho é mostrado.

5.2.2 Instalação das Aplicações Requeridas no RPI

Existem algumas aplicações que são necessárias instalar no Raspberry Pi para conclusão deste projeto. Para o registo de dados, MySQL Apache2 e Phpmyadmin são necessários a instalação, e para o desenvolvimento do *website*, o PHP é necessário.

Mas também, é necessário a instalação do Arduíno IDE no Raspberry Pi, para obter a leitura de valores dos sensores conectados no Arduíno Uno.

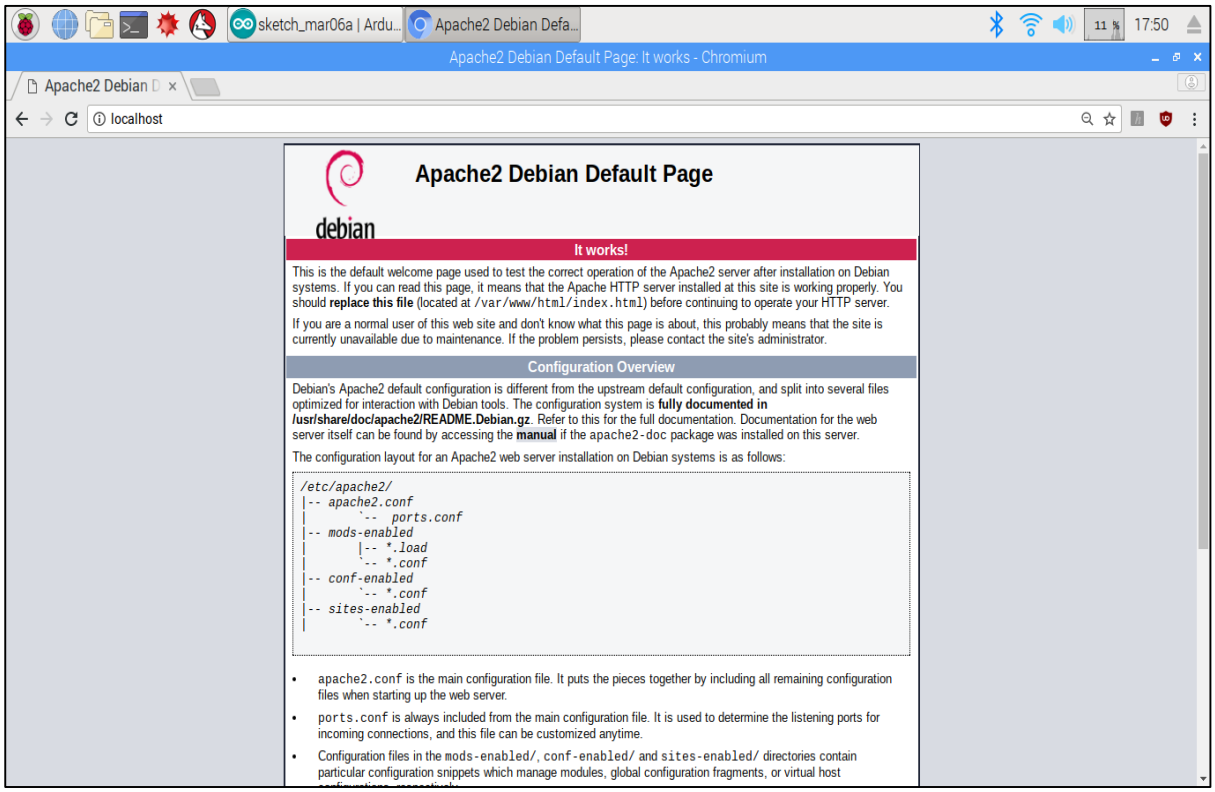
A) Instalação do LAMP (Linux, Apache, MySQL, PHP)

O Rasbian é uma variante Linux oficial do RPI, portanto o Linux é instalado no momento da instalação do SO.

Para a instalação do Apache2, PHP e MySQL, primeiramente deve-se abrir o Terminal LX e digitar os comandos que se seguem na sequência:

1) Primeiro passo, certificar que o RPI está atualizado. Para isso digitar o comando "*sudo apt-get update*" no Terminal LX;
2) Após a atualização, instala-se o Apache2 com o comando "*sudo apt-get install apache2*". Após a instalação do Apache2, por padrão o Apache coloca um arquivo HTML de teste na pasta da WEB do RPI. O arquivo HTML pode ser visualizada entrando com o URL "http://localhost/" no navegador do Raspberry Pi ou "http://192.168.35.108" (seja qual for o endereço IP do RPI);

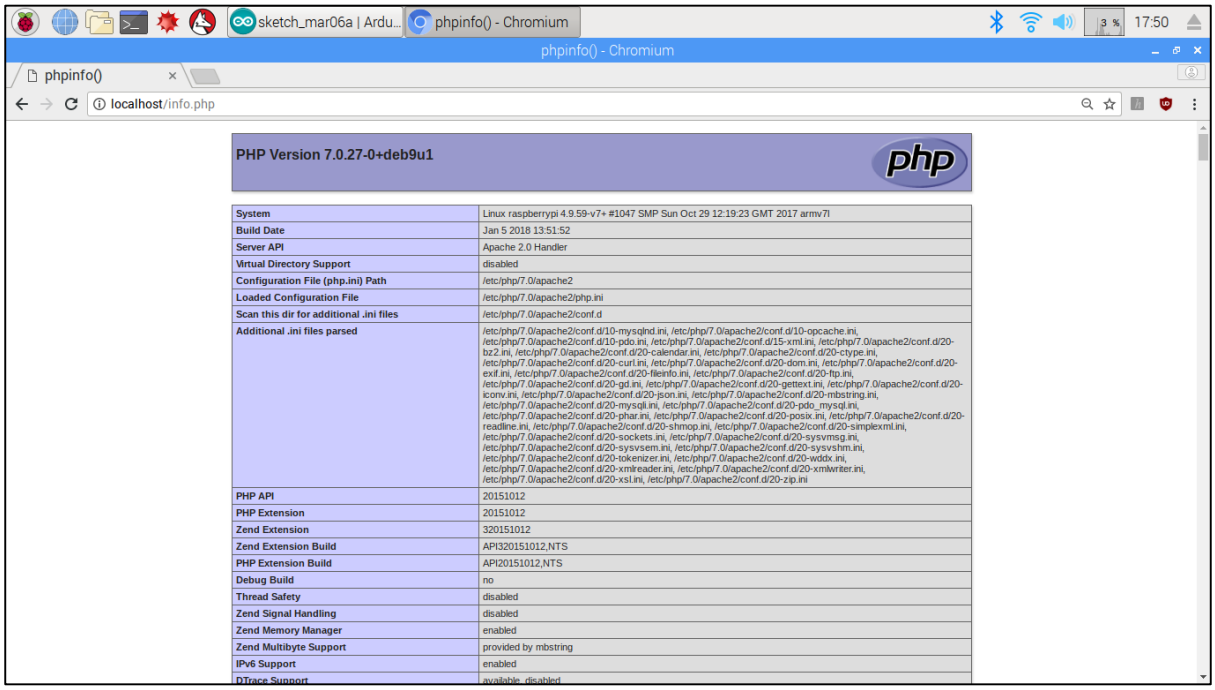
Figura 15: Arquivo HTML de teste do Apache2



Fonte: Elaboração Própria

3) Em seguida instala-se o PHP na versão 7.0, "*sudo apt-get install php7.0*". Depois disso, deve-se acompanhar a instalação do pacote de suporte para a conetividade do banco de dados "*sudo apt-get install php7.0-mysql*";
4) Por fim, instala-se o servidor MySQL, digitando o comando "*sudo apt-get install mysql-server*" e depois "*sudo apt-get install mysql-client*" no Terminal LX. É recomendado atribuir uma senha e esta será a senha do usuário "*root*".

Figura 16: Arquivo teste do PHP7.0



Fonte: Elaboração Própria

B) Instalação do Phpmyadmin

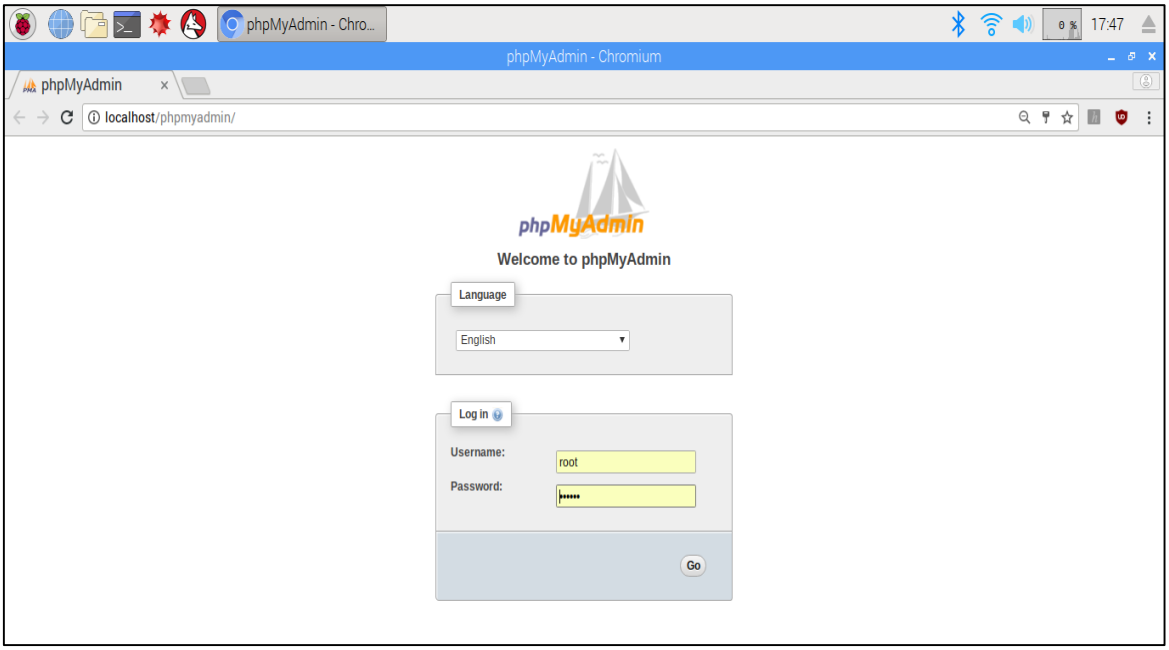
Phpmyadmin é uma interface de rede acessível para administrar bancos de dados MySQL locais e pode tornar os *queries* de banco de dados e gerenciamentos mais simples.

A instalação do phpmyadmin é explicada na seção a seguir em etapas:

1) Instalar o pacote Phpmyadmin digitando o comando, "*sudo apt-get install phpmyadmin*". Durante a instalação, é solicitado para escolher um servidor WEB onde o Apache2 deve ser escolhido;
2) A próxima tarefa é configurar para “dbconfig-common”. Aqui, deve-se selecionar "*Yes*", isso irá pedir-lhe a senha administrativa que é a senha do usuário "*root*" que foi configurada durante a instalação do MySQL na alínea anterior. Em seguida, é solicitado a inserir a senha para "phpmyadmin". A senha para MySQL e para Phpmyadmin é uma coisa diferente, mas pode ser mantida a mesma para ambos. E assim dá-se a conclusão da instalação do phpmyadmin;
3) Para finalizar, é necessário configurar o MySQL para que o usuário "*root*" tenha todos os privilégios com o phpmyadmin. Para tal, digita-se o comando "*sudo –u mysql -p*" no Terminal LX e entrar com a senha atribuída anteriormente. Depois de entrar no

MySQL, deve-se digitar o comando "*GRANT ALL PRIVILEGES on *.* to 'root'@'localhost' IDENTIFIED BY 'password'*".

Figura 17: Phpmyadmin no Navegador



Fonte: Elaboração Própria

Agora, o phpmyadmin está pronto para ser usado. Para isso, basta digitar "*http://localhost/phpmyadmin*" no navegador do RPI ou "*http://192.168.35.108/phpmyadmin*" sob a rede.

C) Instalação do Arduíno IDE

O Arduíno sempre esteve disponível para usuários do Raspberry Pi através do comando "*sudo apt-get install arduino*", mas por se tratar de uma versão antiga não funciona bem com as novas placas Arduíno, pelo que a instalação do Arduíno IDE deve ser feita do seguinte modo:

1) Fazer o download do Arduíno IDE para o processador ARM no *link* https://www.arduino.cc/en/Main/Software;
2) Extrair o arquivo comprimido na pasta de *Downloads* no RPI;
3) Abrir o Terminal LX e digitar "*cd Downloads*", em seguida "*ls*", depois "*cd arduino-nightly*", "*ls*" novamente e escolher "*./install.sh*";
4) Por fim digitar o comando "*sudo ./install.sh*" e a instalação é feita.

5.3 Implementação da Base de Dados

Antes da implementação do banco de dados, é muito importante conhecer todos os requisitos do projeto, sendo os principais são de registar os dados do processo, gerenciá-los e monitorizá-los.

A base de dados, serve em primeiro lugar, para desempenhar o papel de registo de dados. Para o registo de dados, o banco de dados deve ser projetado e implementado de acordo com a estrutura dos dados e informações que precisam ser armazenados e recuperados futuramente.

O MySQL é o servidor de banco de dados, usado com o objetivo de registar os dados do projeto. E phpmyadmin é usado para o gerenciamento do banco de dados, visto que é o administrador de todo o servidor MySQL e, igualmente, responsável por levar MySQL para um navegador, fácilitando assim o gerenciamento.

5.3.1 Criação da Base de Dados

É muito simples criar um novo banco de dados em phpmyadmin. Para isso, necessita-se apenas de executar uma *query* de criação, e o banco de dados aparecerá na lista do banco de dados existentes.

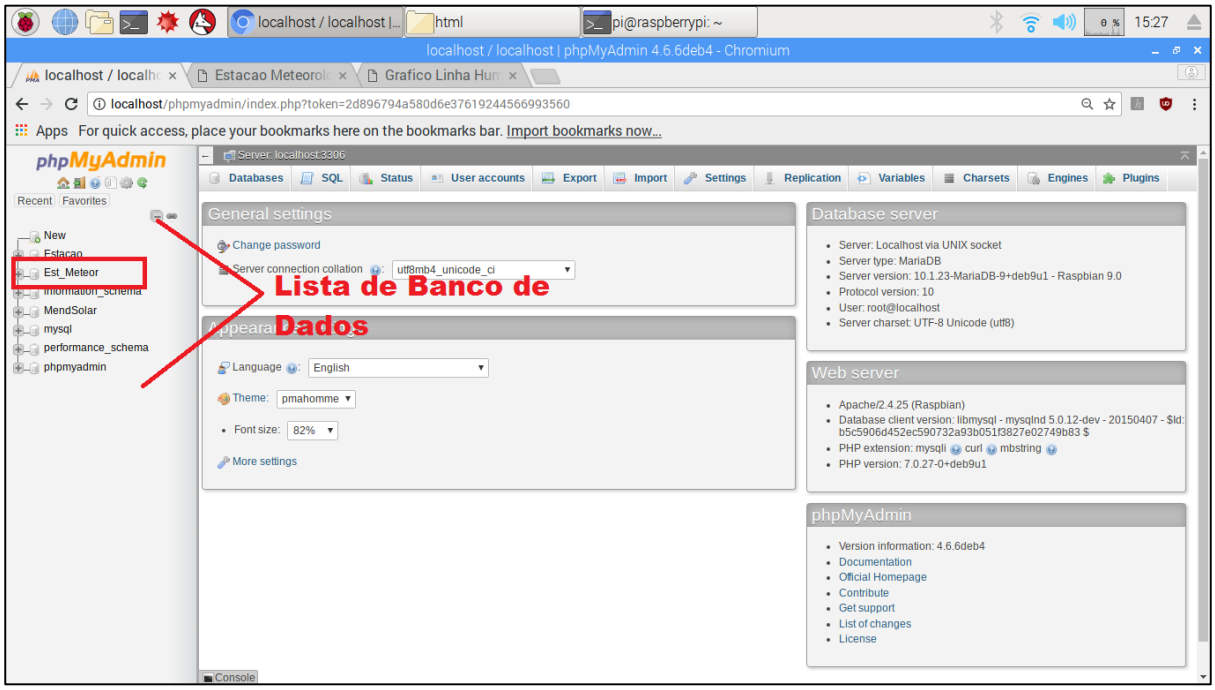
Neste projeto, um banco de dados é criado com o nome como "**Est_Meteor**", que é executado no MySQL *Server*, incorporado no Raspberry Pi.

Para criar o banco de dados basta executar o *query*:

- `CREATE DATABASE Est_Meteor;`

Na lista dos bancos de dados ilustrada na figura 18 (Lista dos Bancos de Dados no Phpmyadmin), interessa para o projeto somente o "**Est_Meteor**", sendo os outros usados para outros fins.

Figura 18: Lista dos Bancos de Dados no Phpmyadmin



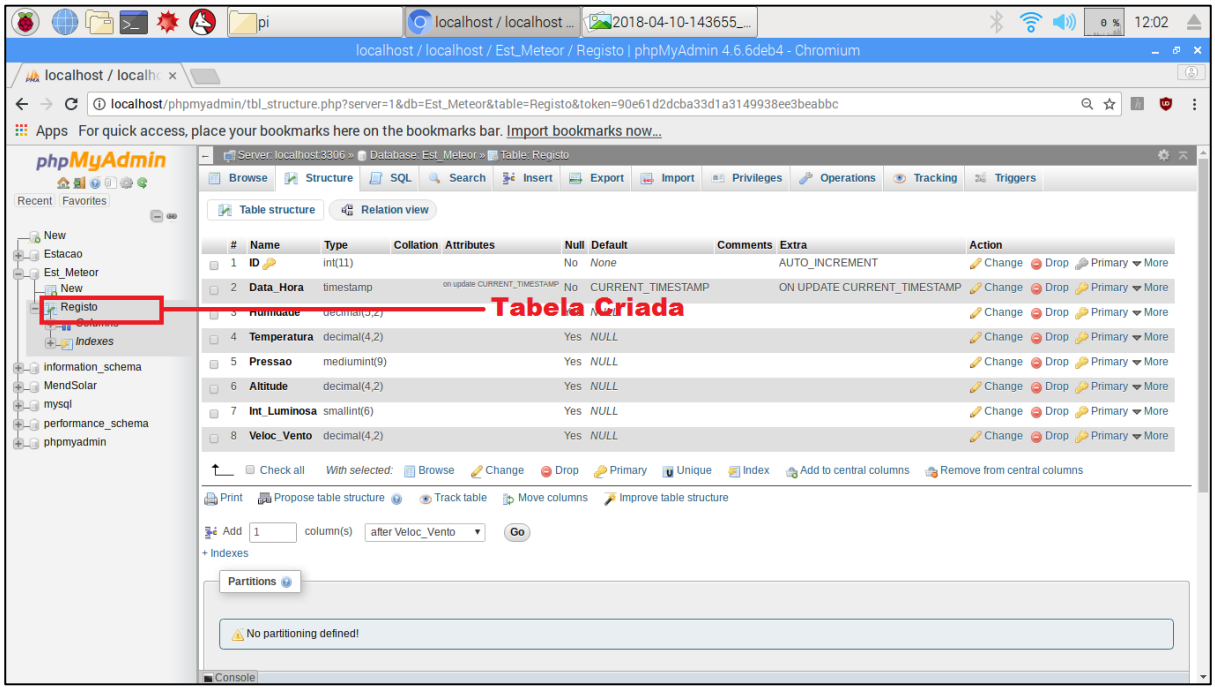
Fonte: Elaboração Própria

5.3.2 Criação da Tabela

Depois de ter criado o banco de dados, agora pode-se criar a tabela. As tabelas também podem ser criadas executando uma *query* de criação. Em baixo segue-se a *query* para criar a tabela:

```
CREATE TABLE `Est_Meteor`.`Registo` (
`ID` INT NOT NULL AUTO_INCREMENT,
`Data_Hora` TIMESTAMP on update CURRENT_TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP,
`Humidade` DECIMAL(5,2) NULL,
`Temperatura` DECIMAL(4,2) NULL,
`Pressao` MEDIUMINT NULL,
`Altitude` DECIMAL(4,2) NULL,
`Int_Luminosa` SMALLINT NULL,
`Veloc_Vento` TINYINT NULL,
PRIMARY KEY (`ID`)) ENGINE = InnoDB;
```

Figura 19: Tabela “Registo” Criada



Fonte: Elaboração Própria

Foi criada somente uma tabela com o nome de “**Registo**”, onde registar-se-á todos os valores de todos os sensores conectados no Arduíno Uno.

5.4 Registo de Dados

O registo de dados é uma das principais tarefas do projeto, que inclui os valores de leitura do processo, usando Arduíno Uno e inserindo os mesmos no banco de dados juntamente com a data e hora no RPI. Não é uma tarefa tão fácil, pois teve algumas coisas a serem feitas e configuradas até fazê-lo funcionar. A configuração que precisa ser feita é explicada em detalhes nas seções a seguir.

5.4.1 Configurações Preliminares

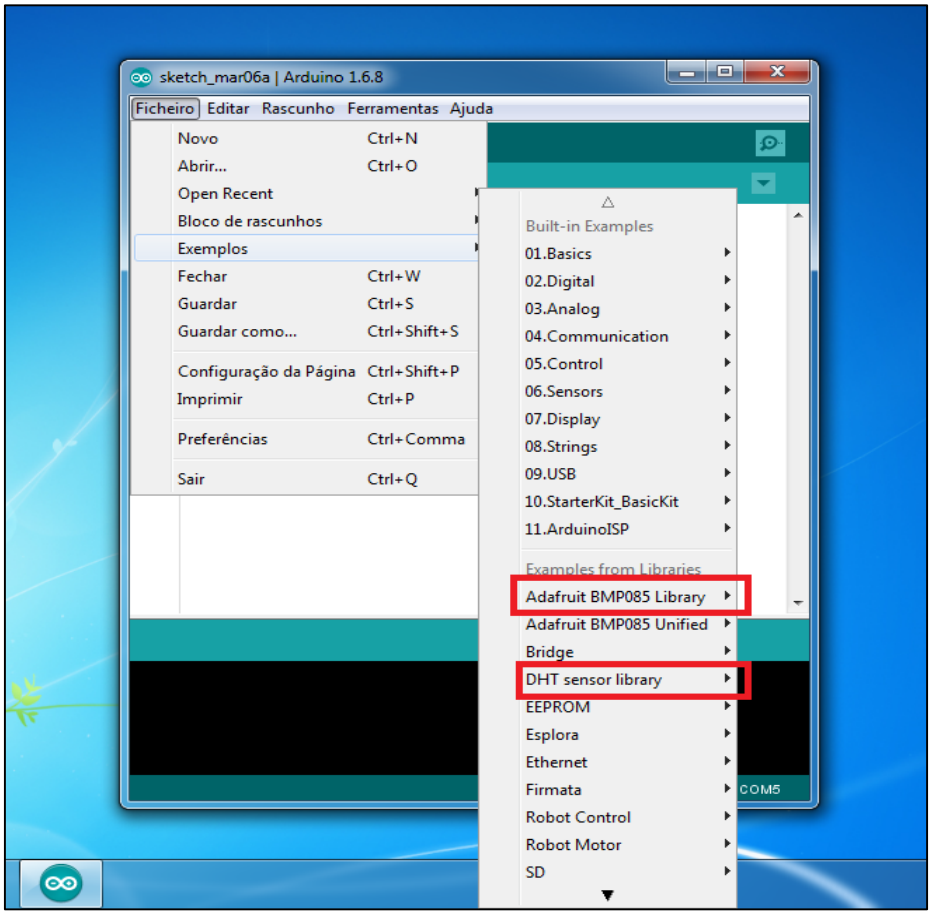
A) Instalação das Bibliotecas Necessárias no Arduíno IDE

Para a leitura dos valores no Arduíno Uno, foi necessário instalar no Arduíno IDE as bibliotecas *"Adafruit BMP085 Library"* referente ao sensor BMP180 e *"DHT Sensor Library"* referente ao sensor DHT22.

As etapas seguintes devem ser seguidas para adicionar essas novas bibliotecas em Arduíno IDE no RPI:

1) *Sketch* > Incluir Biblioteca (*Include Library*) > Gerenciar Bibliotecas (*Manage libraies*)... Ali contém algumas bibliotecas disponível que simplificará a busca por outras bibliotecas.
2) Em seguida, basta digitar os nomes das bibliotecas acima mencionadas na barra de procura e depois clicar em instalar.

Figura 20: Bibliotecas Instaladas no Arduíno IDE



Fonte: Elaboração Própria

B) Instalação das Bibliotecas Necessárias no Python

1) **PySerial Package**

Para que a comunicação *serial* entre o Arduíno Uno e o Raspberry Pi seja possível, é necessário a instalação do pacote PySerial no Raspberry Pi.

No caso de usar o Python 2, deve-se instalar o pacote digitando o comando *"sudo apt-get install python-serial"* no Terminal LX.

No caso de usar o Python 3, que é o caso do projeto, a instalação do pacote é feita digitando o comando *"sudo apt-get install python3-serial"* no Terminal LX.

2) **PyMySQL**

Para conectar o Python a um banco de dados, é preciso um *driver*, que é uma biblioteca usada para interagir com o banco de dados. O PyMySQL é uma biblioteca que tem a finalidade de fazer a conexão entre MySQL e Python3.

Para instalar o PyMySQL deve-se digitar no Terminal LX um dos comandos a seguir:

- *sudo python3 –m pip install pymysql*;
- *sudo apt-get install python-pip > pip install pymysql;*
- *sudo apt-get install python3-pymysql.*

5.4.2 Leitura dos Valores no Arduíno

Feitas as configurações anteriores do sistema de registo, é o momento de codificar o *sketch* (esboço) para a leitura de valores dos sensores na plataforma Arduíno.

Em baixo, segue-se o código em linguagem C++, desenvolvida no Arduíno IDE, com os respetivos comentários de modo a obter a compreensão necessária.

A) Inclusão das bibliotecas necessárias

```
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>
```

B) Definição dos Pinos Utilizados

```
#define DHTPIN 5
#define DHTTYPE DHT22
int sensorPin = A0;
```

C) Atribuição de nomes aos sensores

```
DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085 bmp;
float hum;                  // guarda valores da humidade
float temp;                 // guarda valores da temperatura
int sensorVal = 0;          // guarda valor do LDR

void setup()
{
  Serial.begin(9600);       // iniciar a comunicacao serial
  dht.begin();              // iniciar o sensor DHT22
  if (!bmp.begin()){        //iniciar o sensor BMP180
  Serial.println("BMP180 nao encontrado… Verificar a ligacao!!");
  while (1) {}
  }
}

void loop()
{
    delay(2000);
```

D) Leitura e Impressao dos Valores de DHT22

```
    hum = dht.readHumidity();    // atribuir a funcao na variavel hum
    temp= dht.readTemperature(); // atribuir a funcao na variavel temp
    Serial.print(hum);           // imprime o valor de hum
    Serial.print(" ");           // separar os valores por um espaco
    Serial.print(temp);          // imprime o valor de temp
```

E) Leitura e Impressao dos Valores de BMP180

```
    Serial.print(" ");           // separar os valores por um espaco
    Serial.print(bmp.readPressure()); // imprime o valor de Pressao
    Serial.print(" ");
```

```
    Serial.print(bmp.readAltitude()); // imprime o valor de Altitude
    Serial.print(" ");
```

F) Leitura e Impressao dos Valores de LDR

```
    sensorVal = analogRead(sensorPin);     // leitura de LDR
    Serial.println(sensorVal);             // imprime o valor
```

G) Leitura e Impressao dos Valores de SEN0170

```
    Serial.print(" ");
    int anemVal = analogRead(A1);
    float outvoltage = anemVal * (5.0 / 1023.0);
    int Level = 6*outvoltage;   // A veloc. de vento e proporcional
a tensao de saida
    Serial.println(Level);      // imprime o valor
    delay(10000);
}
```

Obs.: É de extrema importância saber que os códigos para leitura dos sensores DHT22 e BMP180 já se encontravam disponíveis nos exemplos do Arduíno IDE, após a instalação das respetivas bibliotecas. O respetivo código para leitura do sensor SEN0170 encontra-se também disponivel no *site* das informações técnicas do mesmo. Fez-se uma junção e modificação desses códigos de modo a obter num único código a leitura e a impressão *serial* de todos os valores extraídos dos sensores.

5.4.3 Introdução dos Valores no Banco de Dados MySQL

Segue-se o código Python responsável por pegar das saídas dos valores *serial* vindos do Arduíno e envia-los para o banco de dados MySQL, com os respetivos comentários de modo a obter a compreensão da mesma.

A) Importação das bibliotecas

```
import serial
import pymysql
import time
```

B) Estabelecer ligação com o MySQL

```
dbConn = pymysql.connect("localhost","root","123456","Est_Meteor") or
die ("Nao conseguiu conectar ao BD!!")
```

C) Abrir o cursor do banco de dados

```
cursor = dbConn.cursor()
device = '/dev/ttyACM0'    #porta serial q o arduino esta conectado

try:
  print ('Conectando...'),device
  arduino = serial.Serial(device, 9600)
except:
  print ("Falha na conexao!!"),device

try:
  data = arduino.readline()      #leitura de valores do arduino
  pieces = data.split()          #divide os dados por coluna
```

D) Inserir valores no banco de dados

```
  try:
    cursor.execute("INSERT INTO Registo
(Humidade,Temperatura,Pressao,Altitude,Int_Luminosa,Veloc_Vento)
VALUES (%s,%s,%s,%s,%s,%s)",
(pieces[0],pieces[1],pieces[2],pieces[3],pieces[4],pieces[5]))

    dbConn.commit()
    cursor.close()

    except pymysql.IntegrityError:
    print ("falha na insercao de dados!!")

finally:
    cursor.close()
    print ('Dados inseridos com sucesso!!')

except:
    print ("Falha na obtencao de dados do Arduino!")
```

Obs.: Nota-se que o código python acima foi baseado em códigos de outros projetos já desenvolvidos, bem como também as informações obtidas no próprio *site* oficial do MySQL. O código foi estudado, entendido e modificado por forma a atender as necessidades exigidas pelo projeto.

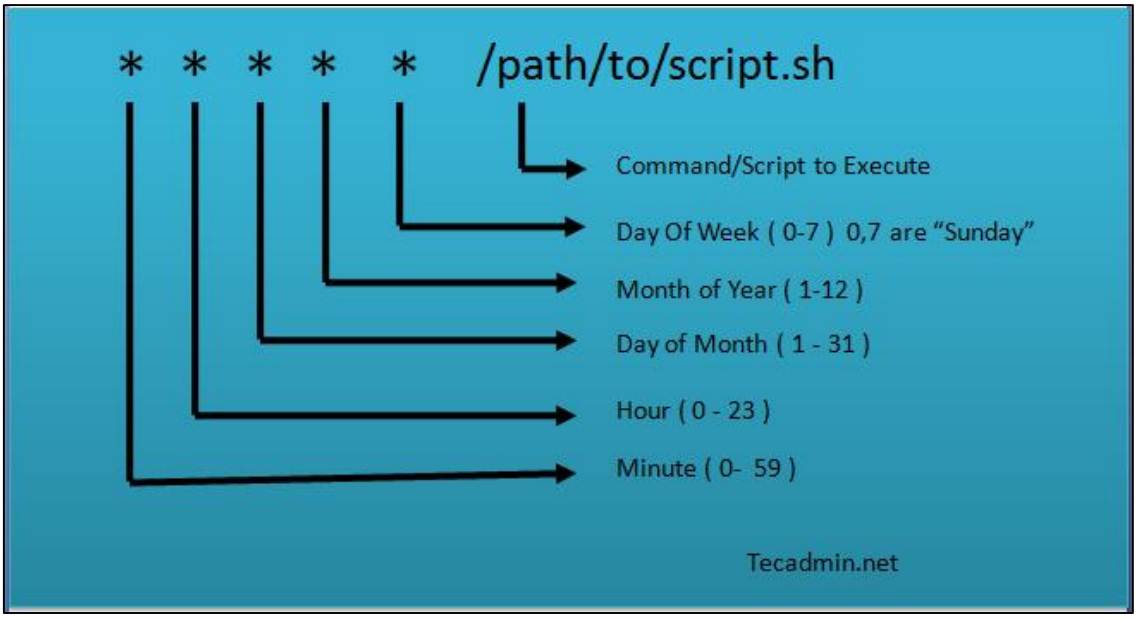
5.4.4 Comando Cron

Cron é um comando utilizado para agendar tarefas recorrentes ou seja, tarefas que repetem-se em intervalos regulares. Essas tarefas denominam-se de Cronjobs que são gerenciados pelo uso de uma tabela chamada Crontab.

O Crontab contém uma lista de comandos que estão programados para serem executados em intervalos de tempo regulares no sistema do seu computador. O comando "*crontab –e*" abre o crontab para edição e permite adicionar, remover ou modificar tarefas agendadas.

A figura a seguir mostra a sintaxe do Crontab:

Figura 21: Sintaxe do Crontab



Fonte: https://tecadmin.net/crontab-in-linux-with-20-examples-of-cron-schedule/

O Crontab é utilizado para executar o código python a cada 10 minutos, ou seja o código python envia dados dos sensores para o banco de dados MySQL a cada 10 minutos.

Para instalar/configurar um novo crontab deve-se ir ao Terminal LX e digitar o comando *"sudo crontab -e"*. Em seguida descer até ultima linha e adicionar uma linha Crontab:

- ** /10 * * * * python3 /home/pi/Desktop/Final_Project/Project.py &*

Nota-se que o "*/home/pi/Desktop/ Final_Project/Project.py*" é o diretório e nome do arquivo python, "*&*" faz com que o programa execute em *background*, e "*python3*" porque utilizou-se o Python na versão 3.5.3.

5.5 Monitoramento e Gerenciamento de Dados

Neste projeto, os dados obtidos pelos sensores são guardados no banco de dados que é monitorado e gerenciado com o uso de um *website*. O *website* é considerado como o sistema de monitoramento e é desenvolvido no Raspberry Pi.

Para o monitoramento de dados, o projeto deve ser capaz de apresentar dados para o visualizador de uma forma fácil e compreensível. Neste sistema, todos os dados são exibidos em medidores, gráficos e tabela de dados.

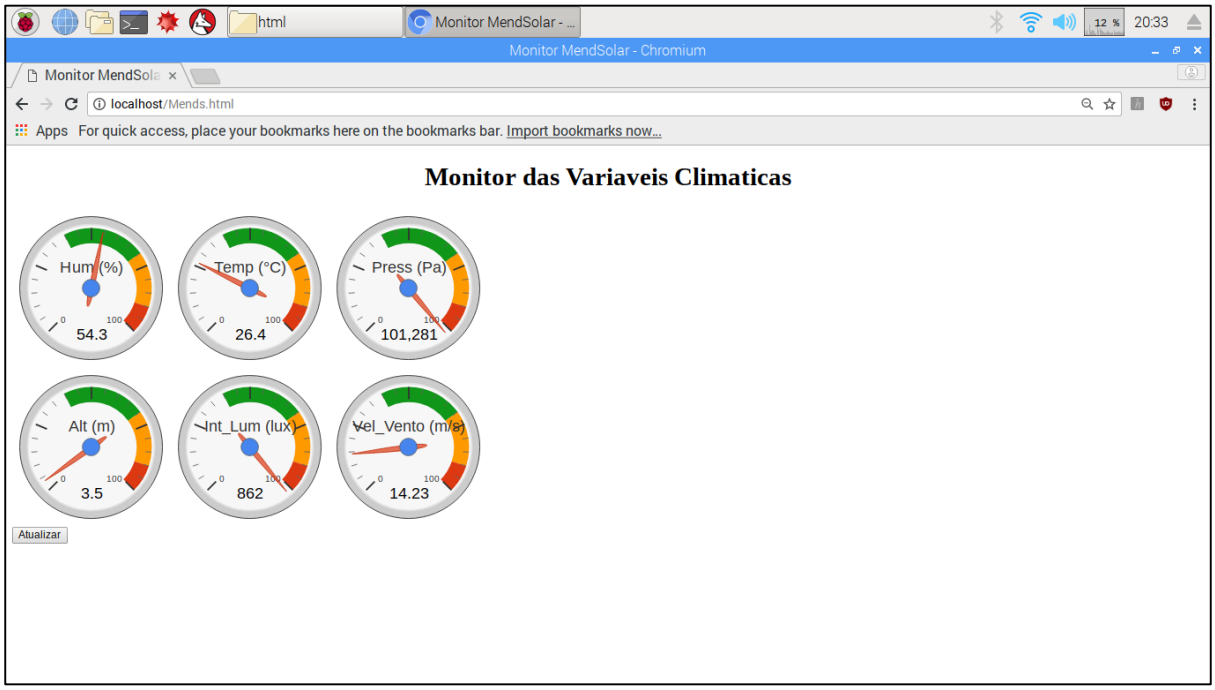
Nota: A conceção e programação do *website* também pode ser feita num PC e, após a conclusão é copiada para o Raspberry Pi. O RPI suporta todas as plataformas e linguagens de programação utilizadas para o desenvolvimento do *website*. Pois, é de maior facilidade e eficácia desenvolver o mesmo num PC, visto que é mais rápido em comparação com o Raspberry Pi.

5.5.1 Estruturação do Website

O *website* dispõe de uma única página onde primeiramente pode-se observar os medidores que exibem, em tempo real, os valores de cada variável climática, vindos dos sensores. Após os medidores vê-se os gráficos de algumas das variáveis climáticas, que indicam as suas respetivas variações ao longo do tempo. Por fim encontra-se uma tabela onde mostra-se os valores registados no banco de dados com as respetivas datas e horas.

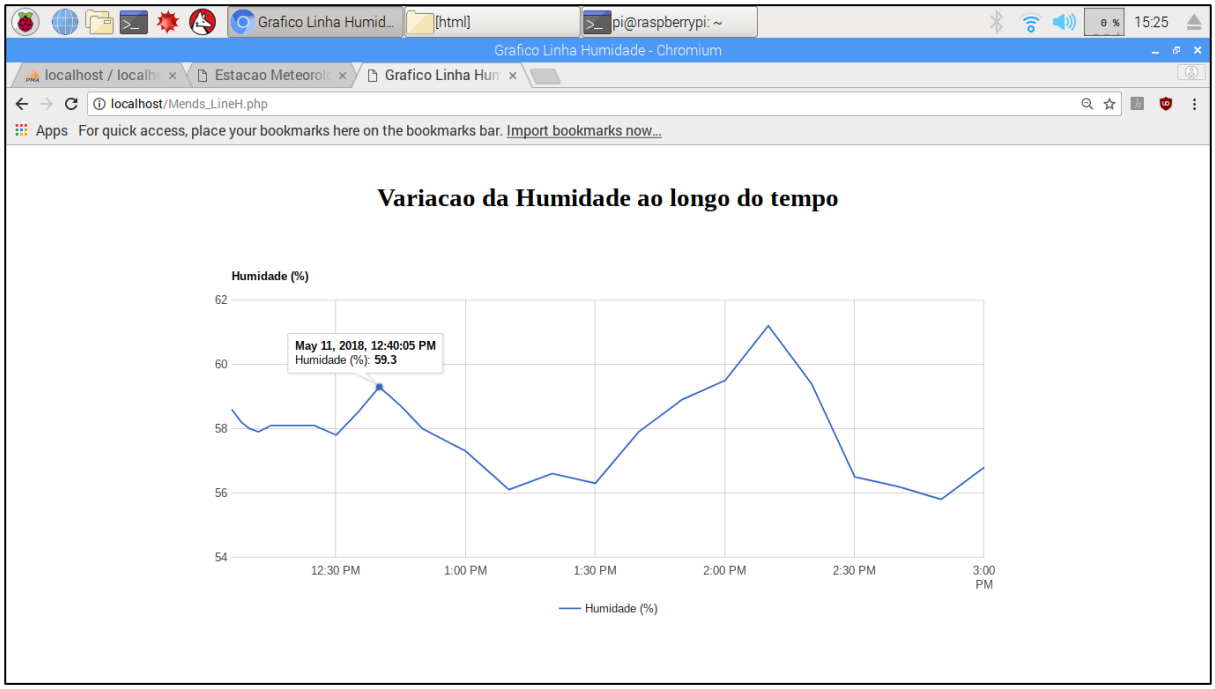
Algumas partes do *website* projetado são mostradas nas figuras a seguir. A figura 22 mostra os medidores das variáveis climáticas em tempo real, figura 23 mostra um dos gráficos de variação de uma variável climática ao longo do tempo e figura 24 mostra a tabela de registo de dados.

Figura 22: Medidores das variáveis climáticas em tempo real



Fonte: Elaboração Própria

Figura 23: Variação da humidade ao longo do tempo



Fonte: Elaboração Própria

Figura 24: Tabela de Registo de Dados

ID	Humidade	Temperatura	Pressao	Altitude	Int_Luminosa	Veloc_Vento	Data/Hora
1	58.60	23.60	101682	-15.03	753	2	2018-05-11 12:05:54
2	58.20	23.70	101657	-13.04	760	4	2018-05-11 12:08:09
3	58.00	23.70	101668	-14.53	756	1	2018-05-11 12:10:03
4	57.90	23.60	101651	-13.71	755	5	2018-05-11 12:12:07
5	58.10	23.60	101675	-14.12	775	4	2018-05-11 12:15:03
6	58.10	23.60	101656	-14.12	770	5	2018-05-11 12:20:03
7	58.10	23.50	101654	-12.79	777	0	2018-05-11 12:25:03
							2018-05-

Fonte: Elaboração Própria

5.5.2 Programação do Website

Para a programação do *website* usou-se o PHP juntamente com as aplicações do Google Charts (Line e Gauge). Dentro das aplicações do Google Charts contêm outras linguagens como HTML, JavaScript, Ajax, JSON e outros mais.

Usou-se também a linguagem CSS para a formatação da tabela de registo de dados, de modo que a sua visulização da mesma seja mais atraente.

Trabalhando com o PHP e o Google Charts juntos, faz com que o *website* tenha uma boa funcionalidade, atendendo as necessidades propostas, e por fim tornando-o assim mais eficaz e atraente.

Nota: Vale destacar que toda programação do *website* não foi desenvolvida pelo autor do presente projeto. A programação foi baseada em outros projetos extraídos da internet, de forma separada, em seguida estudada, entendida e modificada. Por fim fez-se uma junção de cada código por forma a obter o *website*, no qual atende todos objetivos propostos pelo projeto.

5.5.3 Desenvolvimento do Website

Para o desenvolvimento do *website*, os códigos são todos desenvolvidos de forma separada e no fim acoplados num único código. Os códigos são digitados no Terminal LX do RPI acompanhados com o comando *"sudo nano"* com o respetivo diretório, nome e tipo do arquivo, como mostra no exemplo:

- *sudo nano /var/www/html/Project.php*

Para visualizar o resultado do código escrito dirige-se ao navegador do RPI e entrar com o *link* "*http://localhost/Project.php*", por exemplo.

Como já mencionado, o *website* encontra-se divido em 3 partes:

- Visualização dos medidores;
- Visualização dos gráficos;
- Visualização da tabela de registo de dados.

A) Visualização dos medidores

Para a visualização dos medidores criou-se um arquivo PHP, à parte, que tem a finalidade de buscar as informações guardadas na tabela do banco de dados, criado anteriormente. Esse arquivo dipões de duas opções: a primeira busca sempre os últimos dados guardados na tabela e a segunda busca todos os dados guardados na tabela. Para o caso de medidores interessa somente a primeira opção, sendo que a segunda pode ser utilizada na construção de gráficos por exemplo.

As opções são escolhidas consoante o valor de **q**: se **q** for igual a **1** escolhe a primeria opção, senão por defeito escolhe a segunda opção de acordo com o comando *"switch($_GET['q']){"*.

1ª Opção: Busca os últimos dados

```
          case 1:
              $statement=$pdo->prepare("SELECT
Humidade,Temperatura,Pressao,Altitude,Int_Luminosa,Veloc_Vento FROM
Registo ORDER BY ID DESC LIMIT 0,1");
```

```
                $statement->execute();
                $results=$statement->fetchAll(PDO::FETCH_ASSOC);
                $json=json_encode($results);
                echo $json;
            break;
```

2ª Opção: Busca todos os dados

```
            default:
                $statement=$pdo->prepare("SELECT
Humidade,Temperatura,Pressao,Altitude,Int_Luminosa, FROM Registo
ORDER BY ID ASC");
                $statement->execute();
                $results=$statement->fetchAll(PDO::FETCH_ASSOC);
                $json=json_encode($results);
                echo $json;
            break;
```

Após a criação do arquivo PHP, cria-se a seguir um outro arquivo PHP que contém o código do Google Charts Gauge juntamente com o PHP. Esse código busca o resultado da primeira opção do arquivo PHP anteriormente criado e exibe esses dados em medidores no navegador em tempo real. Ou seja, toda vez que for enviada dados para o banco de dados, os medidores exibem o valor do último registo.

B) Visualização dos gráficos

A visualização dos gráficos conta-se com três gráficos de variáveis diferentes (temperatura, humidade e velocidade de vento). Para a exibição desses gráficos é necessário criar um arquivo PHP com 3 códigos diferentes, ou seja, cada gráfico contém um código específico referente a sua variável. Esses códigos são compostos por códigos do Google Charts Line juntamente com PHP, que consistem em pegar todos dados guardados na tabela do banco de dados e criar gráficos de linhas onde mostram a variação das variáveis climáticas escolhidas ao longo do tempo.

C) Visualização da tabela de registo de dados

A última parte, consiste na apresentação de uma tabela de registo de dados. Para exibição dessa tabela cria-se um arquivo PHP que contém um código em PHP com finalidade de exibir a tabela do banco de dados no navegador. Para dar a tabela um aspeto mais atraente utilizou-se um

código CSS. O código CSS é digitado dentro de um arquivo css, que por sua vez está contido numa pasta "*css*". A pasta "*css*" deve situar juntamente com o arquivo PHP, que contém o código da tabela. O código CSS é mostrado a seguir:

```
table tr td{
     padding: 10px;
     text-align: center;
}
```

Obs.: A programação completa do *website* encontra-se disponível nos anexos.

5.6 Sistema do Circuito Eletrónico

No capítulo 3, foi realizado o estudo dos componentes que atenderam as variáveis climáticas necessárias para a criação da Estação Meteorológica. Após esse estudo, pretende-se mostrar a integração de todos os componentes que forma o circuito eletrónico final de toda a estação.

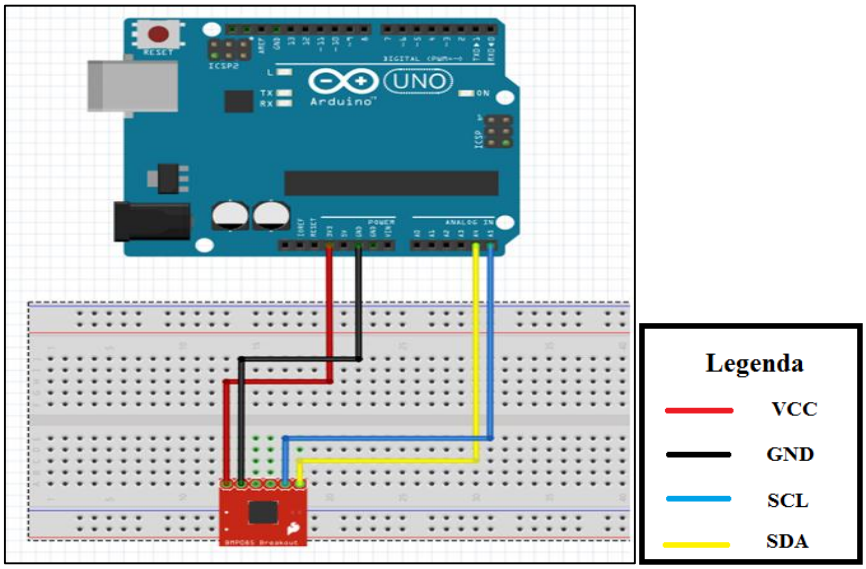
Pode-se observar a seguir como cada sensor é ligada ao microcontrolador Arduíno Uno, especificando quais são os terminais de dados e os de alimentação, formando ao final de todo o proceso a inteligação de todos os componentes.

Todo este processo será mostrado através de imagens do Fritizing, um software auxiliar utilizado para exibição das interligações.

5.6.1 Interligação do Sensor BMP180

De acordo com o estudo no capítulo 3 anterior, o sensor BMP180 possui interface I2C para aquisição de dados, através dos pinos SDL e SDA, ou seja não necessita de componentes externos para funcionar. Assim, as ligações deste sensor com o Arduíno Uno é feita de forma direta como mostra na figura a seguir.

Figura 25: Interligação do BMP180 com o Arduino Uno

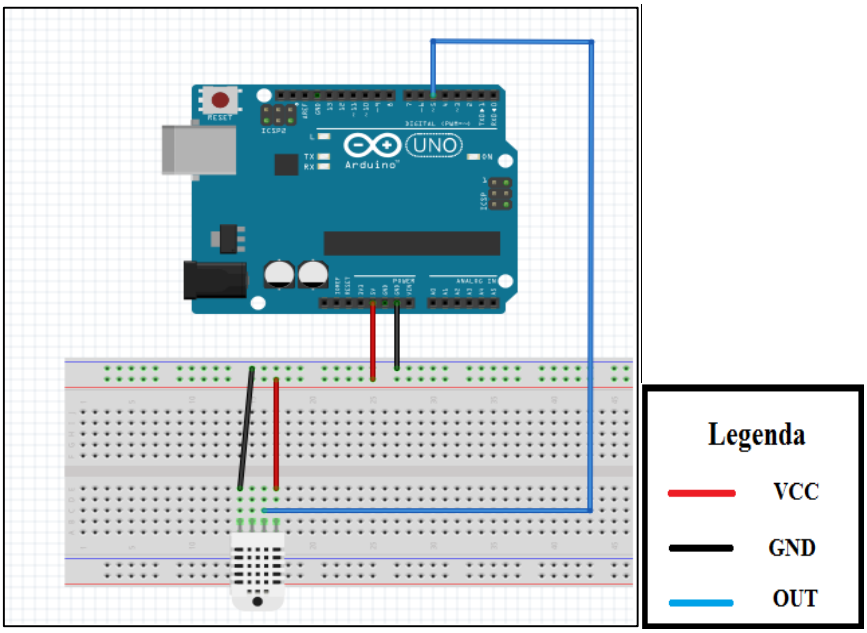


Fonte: Elaboração Própria

5.6.2 Interligação do Sensor DHT22

O sensor DHT22 obtem dados através do protocolo *1-Wire*, retornado o valor aquisitado entre as diferenças dos bulbos digitalmente através do pino 2 do sensor.

Figura 26: Interligação do DHT22 com o Arduino Uno

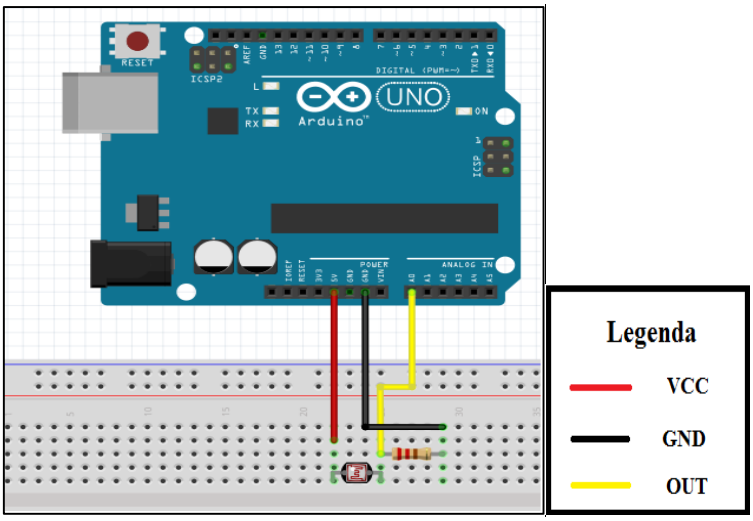


Fonte: Elaboração Própria

5.6.3 Interligação do LDR

O LDR é utilizado no projeto juntamente com uma resistência de10k Ohm, pois o circuito constará apenas de um divisor de tensão. Assim, a ideia é criar uma lógica para ler o valor da resistência do LDR que sua vez estará simulando um verdadeiro sensor de intensidade luminosa cujos valores são expressos em lux em vez de ohm.

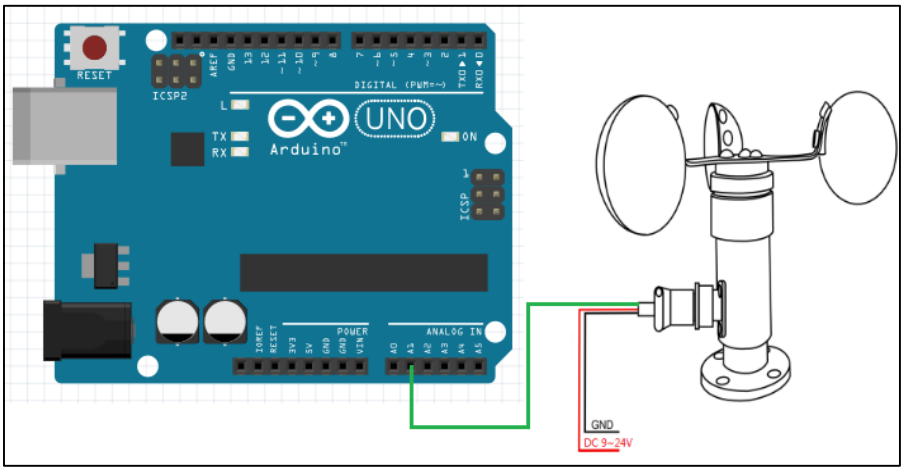
Figura 27: Interligação do LDR com o Arduino Uno



Fonte: Elaboração Própria

5.6.4 Interligação do SEN0170

Figura 28: Interligação do SEN0170 com o Arduino Uno

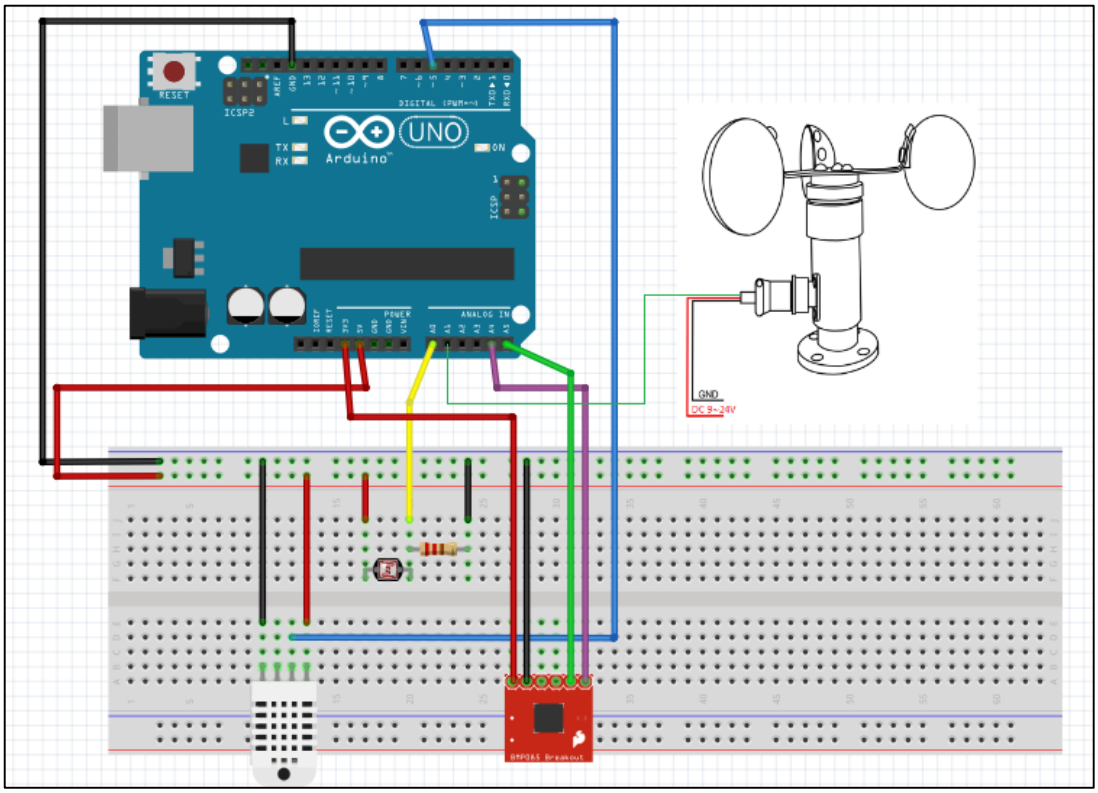


Fonte: Elaboração Própria

5.6.5 Interligação Completa da Estação Metereologica

A figura 29 apresenta a interligação de todos os componentes usados na construção da estação metereológica.

Figura 29: Interligação Completa da Estação Metereologica



Fonte: Elaboração Própria

VI. CONCLUSÃO

6.1 Análise do Projeto Desenvolvido

Pode-se afirmar que o projeto foi concluído dentro do tempo estimado.

Cumpriu-se com os objetivos propostos inicialmente, e todos os problemas encontrados ao longo do seu desenvolvimento foram solucionados, ou pelo menos reduzidos.

Ao final do projeto, a estação consegue-se realizar medições da temperatura ambiente, humidade relativa do ar, pressão atmosférica, altitude (a altitude indicada corresponde à altitude na Atmosfera Padrão) e velocidade de vento, sendo que a intensidade luminosa foi apenas uma simulação com o LDR.

O sistema encontra-se pronto para ser implantado, mas ainda necessita de pequenos ajustes, por exemplo, o *website* precisa fornecer informações mais detalhadas do que um simples relatório (medidores, gráficos e tabela) se for realmente usado na área meteorológica. Os estudos estatísticos da área da implementação são fundamentais e um sistema que gerencie e forneça tais informações é de grande utilidade.

A base do projeto foi o desenvolvimento de uma solução utilizando o microprocessador Arduino Uno e o microcomputador RPI 3B e o objetivo foi cumprido. O sistema de monitoração da estação meteorológica funciona, mas pode ser expandido e aperfeiçoado. É possível desenvolver inúmeros projetos com o RPI, pois trabalhar com o mesmo é prático e relativamente simples.

O projeto apresentou-se viável financeiramente, pois o custo para produção é relativamente baixo levando-se em consideração, mão de obra para o desenvolvimento, programas utilizados e os demais componentes como o microprocessador, microcomputador, dispositivos externos e sensores.

A construção do protótipo proposto neste projeto foi possível graças a alguns conhecimentos adquiridos durante todo o curso de Engenharia em Energias renováveis. Foram aplicados

conceitos de Automação e Controlo, Eletrônica I e II, entre outros. Mas também foi necessário ir muito além dos conhecimentos obtidos durante o curso para aprofundamento em conhecimentos mais específicos para construção do projeto.

Vale destacar a importância das páginas de fórum do Raspberry Pi e do Arduíno, pois nessas páginas foram encontradas soluções de várias configurações que tiveram de ser feitas no RPI para que o projeto fosse finalizado com sucesso.

6.2 Trabalhos Futuros

Como propostas de trabalhos futuros em desenvolvimento de *hardware*, pode-se citar a adição de novos sensores como por exemplo a direção do vento, inclusão de um luminometro (sensor de luminosidade e a radiação incidente), pluviómetro (sensor de chuva), etc.

Também como continuação deste projeto, fica em aberto o estudo e a expansão do *website*, por forma a torna-la mais complexa, mais dinámica e mais interativa com o cliente, pois é um *website* simples que só permite a visualização.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Emerson. **Banco de Dados MySQL e PostgreSQL**. Disponível em: https://www.infowester.com/postgremysql.php, Acesso em 21-12-2017.

ALVIM, G. P. & DI MARCO, P. C., 2016. **Projeto e Construção de uma Estação Meteorológica aplicada a uma Embarcação Teleoperada**. Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro.

arduino.cc. **Arduino Uno Rev3.** Disponível em: https://store.arduino.cc/usa/arduino-uno-rev3, Acesso em 10-12-2017.

arduino.cc. **What Is Arduino?.** Disponível em: http://arduino.cc/en/guide/introduction, Acesso 10-12-2017.

BASTOS, A., 2002. **Instrumentação, Eletrônica Analógica e Digital**. Rio de Janeiro, 360p.

CASTELA, Rodrigo Tenedini. **Introdução a Linguagem PHP**. Disponível em: http://www.dotsharp.com.br/programacao/php/introducao-a-linguagem-php.html, Acesso em 21-12-2017.

create.arduino.cc. **How To Interface Arduino With Raspberrypi**. Disponivel em: https://create.arduino.cc/projecthub/ruchir1674/how-to-interface-arduino-with-raspberrypi-504b06, Acesso 10-12-2017.

dev.mysql.com. **Chapter 7 MySQL Connector/Python Developer Guide**. Disponivel em: https://dev.mysql.com/doc/connectors/en/connector-python.html, Acesso em 15-02-2018.

developers.google.com. **Line Chart**. Disponível em: https://developers.google.com/chart/interactive/docs/gallery/linechart, Acesso em 20-03-2018.

developers.google.com. **Visualization: Gauge**. Disponível em: https://developers.google.com/chart/interactive/docs/gallery/gauge, Acesso em 20-03-2018.

GOMES, Alexandre Guedes, 2015/2016. **Estação Meteorológica com Acesso Remoto**. Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Bragança, Portugal.

GRAÇA, A., 2013. **Introdução a Investigação Científica, Guia Para Investigar e Redigir**. Universidade do Mindelo.

Informações sobre o Google Charts. Disponível em: https://www.schoolofnet.com/curso-graficos-poderosos-com-google-chart-tools/, Acesso em 22-12-2017.

Informações sobre o sensor BMP180. Disponível em: http://cdn.sparkfun.com/datasheets/Sensors/Pressure/BMP180.pdf, Acesso em 21-12-2017.

Informações sobre o sensor DHT22. Disponível em: https://www.adafruit.com/datasheets/Digital%20humidity%20and%20temperature%20sensor%20AM2302.pdf, Acesso em 21-12-2017.

Informações sobre o sensor LDR. Disponível em: http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/curso-de-eletronica-o-que-e-um-ldr-light-dependent-resistor/, Acesso em 26-02-2018.

Informações sobre o sensor SEN0170. Disponível em: https://www.dfrobot.com/wiki/index.php/Wind_Speed_Sensor_Voltage_Type(0-5V)_SKU:SEN0170, Acesso em 22-12-2017.

Informações sobre o software Fritzing. Disponível em: http://fritzing.org/home/, Acesso em 13-04-2018.

instructables.com. **Interface Arduino To Mysql Using Python.** Disponivel em: http://www.instructables.com/id/Interface-Arduino-to-MySQL-using-Python/, Acesso em 15-02-2018.

leanpub.com. **Raspberry Pi: Measure, Record, Explore.** Disponível em: https://leanpub.com/RPiMRE/read#leanpub-auto-web-server-and-php, Acesso em 17-01-2018.

MATTOS, Charles Saad, 2005. **Sistema de Monitoração de Estação Meteorológica.** Centro Universitário Positivo (UNICENP), Curitiba.

MOREIRA, Lucia, 2002. **Sensores de Temperatura Princípios e Aplicações**. Ed. Help Temperatura & Metrologia Treinamento e Consultoria.

mundoclima.com.br. **Como funciona e quais as vantagens da estação meteorológica?.** Disponível em: http://blog.mundoclima.com.br/como-funciona-e-quais-as-vantagens-da-estacao-meteorologica/, Acesso em 9-12-2017.

NITERÓI, 2009. **Tutorial de Introdução ao Python.** Universidade Federal Fluminense - Centro Tecnológico, Brasil.

PENIDO, Édilus de C. C. & TRINDADE, Ronaldo Silva, 2013. **Microcontroladores**. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, Brasil.

PRAJAPATI, Kishan, 2015**. Process Control and Monitoring using Arduino and Raspberry Pi**. Telemark University College, Department of Technology, Noruega.

raspberrypi.org. **Raspberry Pi 3 Model B**. Disponível em: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, Acesso em 15-01-2018.

raspberrypi.org. **Raspberry Pi Software Guide**. Disponível em: https://www.raspberrypi.org/learning/software-guide/quickstart/, Acesso em 17-01-2018.

raspberrypi.org. **Setting Up An Apache Web Server On A Raspberry Pi**. Disponível em: https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md, Acesso em 17-01-2018.

ROSA, Edson Bruno Lara & CROCE, José António Garcia, 2016. **Construção De Uma Estação Meteorologia De Baixo Custo**. Instituto Federal de Educação, Ciência e Tecnologia, São Paulo.

tutorialspoint.com. **Python 3: MySQL Database Access.** Disponivel em: https://www.tutorialspoint.com/python3/python_database_access.htm, Acesso em 15-02-2018.

VASCONCELOS, Adriano. **O que é PHP?.** Disponivel em: https://secaoweb.com.br/blog/php-o-que-e-e-quais-suas-caracteristicas/, Acesso em 21-10-2017.

VASCONCELOS, José Braga de, 2015. **Python: Algoritmia e Programação Web, FCA**. Editora de informática.

WATANABE, Ana T. Y., 2011. **O UNIVERSO HCS08QG8, Teoria, Linguagem Assembly, atividades de laboratório e projetos**.

wattnotions.com. **Connecting Sensors To The Internet Using An Arduino And A Raspberry Pi.** Disponivel em: http://www.wattnotions.com/connecting-sensors-to-the-internet-using-an-arduino-and-a-raspberry-pi/, Acesso em 05-02-2018.

webslesson.info. **How to Make Google Line Chart by Using PHP JSON Data.** Disponível em: http://www.webslesson.info/2017/08/how-to-make-google-line-chart-by-using-php-json-data.html, Acesso em 26-03-2018.

youtube.com. **Medidor en Tiempo Real con Javascript, Php, Mysql y Google Charts, sensor humedad y temperatura.** Disponível em: https://www.youtube.com/watch?v=WyUM--RGLH0, Acesso em 19-03-2018.

VIII. ANEXOS

Anexo A: Código C++ do Arduino IDE

```
// Inclusao das bibliotecas necessárias

#include <DHT.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>

// Definicao dos pinos utilizados

#define DHTPIN 5
#define DHTTYPE DHT22
int sensorPin = A0;

// Atribuicao de nomes aos sensores

DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085 bmp;

// Atribuicao de nomes as variaveis

float hum;
float temp;
int sensorVal = 0;

void setup()
{
  Serial.begin(9600);
  dht.begin();
  if (!bmp.begin()) {
  Serial.println("BMP180 não encontrado… Verificar a
ligacao!!!");
```

```
  while (1) {}
  }
}

void loop()
{
    delay(2000);

    //Leitura e impressao dos valores de DHT

    hum = dht.readHumidity();
    temp= dht.readTemperature();
    Serial.print(hum);
    Serial.print(" ");
    Serial.print(temp);

    // Leitura e impressao dos valores de BMP

    Serial.print(" ");
    Serial.print(bmp.readPressure());
    Serial.print(" ");
    Serial.print(bmp.readAltitude(101500));
    Serial.print(" ");

    // Leitura e impressao dos valores de LDR

    sensorVal = analogRead(sensorPin);
    Serial.print(sensorVal);

    // Leitura e impressao dos valores de SEN0170

    int anemVal = analogRead(A1);
    float outvoltage = anemVal * (5.0 / 1023.0);
    Serial.print(" ");
    int Level = 6*outvoltage;
```

```
    Serial.println(Level);
    delay(10000);
}
```

Anexo B: Código Python

```
import serial
import pymysql
import time

#Estabelecer ligacao com o MySQL

dbConn =
pymysql.connect("localhost","root","123456","Est_Meteor") or
die ("Nao conseguiu conectar ao BD!!")

#Abrir o cursor do Banco de Dados

cursor = dbConn.cursor()

device = '/dev/ttyACM0' #Porta Serie
try:
  print ('Conectando...'),device
  arduino = serial.Serial(device, 9600)
except:
  print ("Falha na conexao!!"),device

try:
  data = arduino.readline()  #leitura de dados do arduino
  pieces = data.split()       #dividir os dados

#Inserir dados no Banco de Dados

  try:
    cursor.execute("INSERT INTO
Registo(Humidade,Temperatura,Pressao,Altitude,Int_Luminosa,Vel
oc_Vento) VALUES (%s,%s,%s,%s,%s,%s)",
(pieces[0],pieces[1],pieces[2],pieces[3],pieces[4],pieces[5]))
```

```
        dbConn.commit()
        cursor.close()  #Fecha o cursor

    except pymysql.IntegrityError:
        print ("falha na insercao de dados!!")

    finally:
        cursor.close()  #Fecha o cursor
        print ('Dados inseridos com sucesso!!')

except:
    print ("Falha na obtencao de dados do Arduino!")
```

Anexo C: Código Geral do Website

C1: Medidores

1. Código do arquivo PHP que busca dados no banco de dados

```
<?php
header('Content-Type: application/json');
$pdo=new
PDO("mysql:dbname=Est_Meteor;host=localhost","root","123456");
switch($_GET['q']){
          // Busca ultimo Dado
          case 1:
              $statement=$pdo->prepare("SELECT
Humidade,Temperatura,Pressao,Altitude,Int_Luminosa,Veloc_Vento
FROM Registo ORDER BY ID DESC LIMIT 0,1");
               $statement->execute();
               $results=$statement-
>fetchAll(PDO::FETCH_ASSOC);
               $json=json_encode($results);
               echo $json;
          break;
          // Busca todos os dados
          default:
               $statement=$pdo->prepare("SELECT
Humidade,Temperatura,Pressao,Altitude,Int_Luminosa,Veloc_Vento
FROM Registo ORDER BY ID ASC");
               $statement->execute();
               $results=$statement-
>fetchAll(PDO::FETCH_ASSOC);
               $json=json_encode($results);
               echo $json;
          break;
}
```

```
?>
```

2. Código dos Medidores

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>
    <title>Estacao Meteorologica</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery
.min.js"></script>
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['gauge']});
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Label', 'Value'],
          ['Hum (%)', 0],
          ['Temp (°C)', 0],
          ['Press (Pa)', 0],
          ['Alt (m)', 0],
          ['Int_Lum (lux)', 0],
          ['Vel_Vento (m/s)', 0]
          ]);
        var options = {
          width: 600, height: 600,
          redFrom: 90, redTo: 100,
          yellowFrom:70, yellowTo: 90,
          greenFrom: 40, greenTo: 70,
          minorTicks: 5
        };
```

```
        var chart = new
google.visualization.Gauge(document.getElementById('Medidores'
));
        chart.draw(data, options);
        setInterval(function() {
            var JSON=$.ajax({
                url:"http://localhost/Mends.php?q=1",
                dataType: 'json',
                async: false}).responseText;
            var Resposta=jQuery.parseJSON(JSON);

          data.setValue(0, 1,Resposta[0].Humidade);
          data.setValue(1, 1,Resposta[0].Temperatura);
          data.setValue(2, 1,Resposta[0].Pressao);
          data.setValue(3, 1,Resposta[0].Altitude);
          data.setValue(4, 1,Resposta[0].Int_Luminosa);
          data.setValue(5, 1,Resposta[0].Veloc_Vento);
          chart.draw(data, options);
          }, 1300);
      }
    </script>
</head>
<body>
     <h1 align="center"><b>Monitor das Variaveis
Climaticas</b></h1>
     <div id="Medidores" ></div>
     <div id="chart_div" style="width: 450px; height:
0px;"></div>
</body>
</html>
```

C2: Gráficos de Variação

```
<?php
```

```
$connect = mysqli_connect("localhost", "root", "123456",
"Est_Meteor");
$query = 'SELECT ID,Temperatura,UNIX_TIMESTAMP(CONCAT_WS(" ",
Data_Hora)) AS datetime FROM Registo ORDER BY Data_Hora DESC';

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Data_Hora',
  'type' => 'datetime'
 ),
 array(
  'label' => 'Temperatura (°C)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
      "v" => 'Date(' . $datetime[0] . '000)'
     );
 $sub_array[] =  array(
      "v" => $row["Temperatura"]
     );
 $rows[] =  array(
     "c" => $sub_array
```

```
    );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
?>


<html>
 <head>
  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
js"></script>
  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawTemChart);
   function drawTemChart()
   {
    var dataT = new google.visualization.DataTable(<?php echo
$jsonTable; ?>);
    var options = {
     title:'Temperatura (°C)',
     legend:{position:'bottom'},
     chartArea:{width:'95%', height:'65%'}
    };
    var chart = new
google.visualization.LineChart(document.getElementById('line_c
hartT'));
    chart.draw(dataT, options);
   }
  </script>
  <style>
  .page-wrapper
```

```
  {
   width:1000px;
   margin:0 auto;
  }
  </style>
 </head>
 <body>
  <div class="page-wrapper">
   <br />
   <h1 align="center">Variacao da Temperatura ao longo do
tempo</h1>
   <div id="line_chartT" style="width: 100%; height:
500px"></div>
  </div>
 </body>
</html>
```

```
<?php
$connect = mysqli_connect("localhost", "root", "123456",
"Est_Meteor");
$query = 'SELECT ID,Humidade,UNIX_TIMESTAMP(CONCAT_WS(" ",
Data_Hora)) AS datetime FROM Registo ORDER BY Data_Hora DESC';

$result = mysqli_query($connect, $query);
$rows = array();
$table = array();

$table['cols'] = array(
 array(
  'label' => 'Data_Hora',
  'type' => 'datetime'
 ),
```

```
 array(
  'label' => 'Humidade (%)',
  'type' => 'number'
 )
);

while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
      "v" => 'Date(' . $datetime[0] . '000)'
     );
 $sub_array[] =  array(
      "v" => $row["Humidade"]
     );
 $rows[] =  array(
     "c" => $sub_array
    );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
?>

<html>
 <head>
  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
js"></script>
  <script type="text/javascript">
```

```
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawHumChart);
   function drawHumChart()
   {
    var dataH = new google.visualization.DataTable(<?php echo
$jsonTable; ?>);
    var options = {
     title:'Humidade (%)',
     legend:{position:'bottom'},
     chartArea:{width:'95%', height:'65%'}
    };
    var chart = new
google.visualization.LineChart(document.getElementById('line_c
hartH'));
    chart.draw(dataH, options);
   }
  </script>
  <style>
  .page-wrapper
  {
   width:1000px;
   margin:0 auto;
  }
  </style>
 </head>
 <body>
  <div class="page-wrapper">
   <br />
   <h1 align="center">Variacao da Humidade ao longo do
tempo</h1>
   <div id="line_chartH" style="width: 100%; height:
500px"></div>
  </div>
```

```
 </body>
</html>



<?php
$connect = mysqli_connect("localhost", "root", "123456",
"Est_Meteor");
$query = 'SELECT ID,Veloc_Vento,UNIX_TIMESTAMP(CONCAT_WS(" ",
Data_Hora)) AS datetime FROM Registo ORDER BY Data_Hora DESC';


$result = mysqli_query($connect, $query);
$rows = array();
$table = array();


$table['cols'] = array(
 array(
  'label' => 'Data_Hora',
  'type' => 'datetime'
 ),
 array(
  'label' => 'Velocidade de Vento (m/s)',
  'type' => 'number'
 )
);


while($row = mysqli_fetch_array($result))
{
 $sub_array = array();
 $datetime = explode(".", $row["datetime"]);
 $sub_array[] =  array(
      "v" => 'Date(' . $datetime[0] . '000)'
     );
```

```
 $sub_array[] =  array(
      "v" => $row["Veloc_Vento"]
     );
 $rows[] =  array(
     "c" => $sub_array
    );
}
$table['rows'] = $rows;
$jsonTable = json_encode($table);
?>


<html>
 <head>
  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
js"></script>
  <script type="text/javascript">
   google.charts.load('current', {'packages':['corechart']});
   google.charts.setOnLoadCallback(drawVentChart);
   function drawVentChart()
   {
    var dataV = new google.visualization.DataTable(<?php echo
$jsonTable; ?>);
    var options = {
     title:'Velocidade de Vento (m/s)',
     legend:{position:'bottom'},
     chartArea:{width:'95%', height:'65%'}
    };
    var chart = new
google.visualization.LineChart(document.getElementById('line_c
hartV'));
```

```
    chart.draw(dataV, options);
   }
  </script>
  <style>
  .page-wrapper
  {
   width:1000px;
   margin:0 auto;
  }
  </style>
 </head>
 <body>
  <div class="page-wrapper">
   <br />
   <h1 align="center">Variacao da Veloc. de Vento ao longo do
tempo</h1>
   <div id="line_chartV" style="width: 100%; height:
500px"></div>
  </div>
 </body>
</html>
```

C3: Tabela de Registo de Dados

```
<html>
        <head>
        <h1 align="center">Tabela de Registo de Dados</h1>
        </head>
</html>
<html>
        <head>
        <link  href="css/test.css" rel="stylesheet">
```

```
        </head>
</html>

<?php
$con = new mysqli("localhost","root","123456","Est_Meteor");
if($con->connect_errno){
echo "Erro ao conectar à base de dados";
}

$sql = "SELECT*FROM Registo" or die(mysqli_error($con));
$record=$con->query($sql);

echo "<table border=1 width=500>
<tr>
     <td colspan=4 BGCOLOR=#2196F3><strong> ID </strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Humidade
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Temperatura
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Pressao
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Altitude
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Int. Luminosa
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Veloc. Vento
</strong></td>
     <td colspan=4 BGCOLOR=#2196F3><strong> Data e Hora
</strong></td>
</tr>";

while($linha=$record->fetch_assoc()){

echo "<tr>
          <td colspan=4>".$linha['ID']."</td>
```

```
            <td colspan=4>".$linha['Humidade']."</td>
            <td colspan=4>".$linha['Temperatura']."</td>
            <td colspan=4>".$linha['Pressao']."</td>
            <td colspan=4>".$linha['Altitude']."</td>
            <td colspan=4>".$linha['Int_Luminosa']."</td>
            <td colspan=4>".$linha['Veloc_Vento']."</td>
            <td colspan=4>".$linha['Data_Hora']."</td>
      </tr>";
}

echo"</table>";
?>
```